***How to improve performance of any module in DM***

Steps to identify the performance problem and fix those:

1. Identify the query which is taking long time to execute in module by DBMS\_profiler

And fix that query if possible.

1. Identify all the triggers of those tables which are targeting in the module

You can use below query with table name as args

select trigger\_name,Table\_name,status from All\_Triggers where table\_name IN (

'GCHD',

'CHDRPF',

'GCHPPF',

'GCHIPF',

'GMHDPF',

'GMHIPF',

);

If any trigger is enabled then please write that code into module with correct logic and disable that trigger before execution of that module and enable after it.

Ex.

ALTER TRIGGER vm1dta.TR\_GCHPPF DISABLE;

ALTER TRIGGER vm1dta.TR\_GMHDPF DISABLE;

1. Identify all the sequence in the module and check the cache size.

Query:

select \* from all\_sequences where sequence\_owner='VM1DTA' and sequence\_name in (

'SEQ\_GCHPPF',

'SEQ\_GMHDPF');

Id cahce size is less than 20 then set cache size as 20 with below query:

ALTER SEQUENCE VM1DTA.SEQ\_CLEXPF CACHE 20;

1. Check main cursor query if possible please join the required tables and get the pre loaded data which can use for validation and insertion, we can check the pre loaded package mapped as well and replace this map by joined tables in main query it self.

Ex.

select A.\*,c.cltdob from vm1dta.titdmgmbrindp1\_p A

INNER JOIN vm1dta.zdclpf\_1412 B

on substr(A.refnum , 1,8)=B.zentity

LEFT OUTER JOIN vm1dta.CLNTPF C

on B.zigvalue = c.clntnum

WHERE A.RECIDXMBINP1 between start\_id and end\_id

ORDER BY LPAD(A.refnum, 15, '0') ASC;

1. Please use the bulk fetch from main cursor with limit clause as per PGA memory.

Ex.

OPEN cur\_mbr\_ind\_p1;

LOOP

FETCH cur\_mbr\_ind\_p1 bulk collect

into mbrind1\_list limit 5000;

1. Please use the bulk insertion for this while iterating we will add table object into list and at end after loop we will insert using forall.

Adding object into list:

gchdindex := gchdindex + 1;

gchd\_list.extend;

gchd\_list(gchdindex) := obj\_gchd;

insert in one shot using forall:

idx := gchd\_list.first;

IF (idx IS NOT NULL) THEN

FORALL idx IN gchd\_list.first .. gchd\_list.last

INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO VM1dta.GCHD VALUES gchd\_list (idx);

END IF;

1. Parallel implementation for each module:

Steps:

1. First create all stage tables in VM1DTA schema and insert the data from stage table using DB links.

Create table:

create table vm1dta.JD\_TITDMP1 as SELECT \* FROM titdmgmbrindp1@database\_link1 where 1=2;

put the required indexes:

create index….

1. Insert the record from stagetable to vm1dta table using below query(Must use the hint append for fast insertion).

Ex.

truncate table vm1dta.JD\_TITDMP1;

INSERT /\*+ APPEND \*/ INTO vm1dta.JD\_TITDMP1 SELECT \* FROM titdmgmbrindp1@database\_link1;

1. Identify the how many source table(stage tales ) are there in your module.

If it is one then no issue go to step D directly

If it has more than one then please add unique column of first table to second as well and update that column with proper relation b/w thow two tables.

Ex. In below query RECIDXMBINDP3 in new column which is added into P2 table as sycn with P1’s RECIDXMBINP1 for parallel use.

UPDATE vm1dta.titdmgmbrindp2\_P t1

SET (t1.RECIDXMBINDP3) = (

SELECT t2.RECIDXMBINP1

FROM TITDMGMBRINDP1@database\_link1 t2

WHERE t1.refnum = t2.refnum)

WHERE EXISTS (

SELECT 1

FROM TITDMGMBRINDP1@database\_link1 t2

WHERE t1.refnum = t2.refnum );

1. Please identify the unique column in your source table for chunks.
2. Once you have identified the column please calculate the number of records in source table and divide that by core number and you will get the chunk size.

Ex.

DBMS\_PARALLEL\_EXECUTE.CREATE\_CHUNKS\_BY\_NUMBER\_COL('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK','VM1DTA','TITDMGMBRINDP1\_P','RECIDXMBINP1',36003);

1. Once you done the with above step please write the parallel execution proc for that module based on below sample.

|  |
| --- |
| CREATE OR REPLACE PROCEDURE dm2\_mbrind\_parallel\_execute\_av (  i\_schedulename IN VARCHAR2,  i\_schedulenumber IN VARCHAR2,  i\_zprvaldyn IN VARCHAR2,  i\_company IN VARCHAR2,  i\_fsucocompany IN VARCHAR2,  i\_usrprf IN VARCHAR2,  i\_transcode IN VARCHAR2,  i\_vrcmtime IN NUMBER,  i\_vrcmuser IN NUMBER,  i\_trdt1 IN NUMBER,  i\_acctyear IN NUMBER,  i\_acctmonth IN NUMBER,  i\_vrcmtermid IN VARCHAR2  )  AUTHID current\_user  AS  l\_sql\_stmt VARCHAR2(2000 CHAR);  l\_sql\_stmt1 VARCHAR2(2000 CHAR);  l\_status NUMBER;  l\_chunk\_stmt VARCHAR2(2000 char);  cont NUMBER;  cont1 NUMBER;  v\_timestart NUMBER := dbms\_utility.get\_time;  BEGIN  SELECT  COUNT(1) into cont1  FROM  vm1dta.titdmgmbrindp2\_p;  IF ( cont1 > 0 ) THEN  SELECT  COUNT(\*)  INTO cont  FROM  user\_parallel\_execute\_tasks  WHERE  task\_name = 'DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK';  IF cont > 0 THEN  dbms\_parallel\_execute.drop\_task('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK');  END IF;  dbms\_parallel\_execute.create\_task('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK');  dbms\_parallel\_execute.create\_chunks\_by\_number\_col('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK', 'VM1DTA', 'TITDMGMBRINDP1\_P', 'RECIDXMBINP1'  , 36003);  /\* l\_chunk\_stmt:= 'SELECT start\_id, end\_id FROM TITDMGMBRINDP1\_CHUNKS';  DBMS\_PARALLEL\_EXECUTE.CREATE\_CHUNKS\_BY\_SQL (  task\_name => 'UPDATE\_TITDMGMBRINDP1\_MERGE'  , sql\_stmt => l\_chunk\_stmt  , by\_rowid => FALSE  ); \*/  /\*l\_sql\_stmt := 'MERGE INTO TITDMGMBRINDP1@stagedblink a USING  (select \* from TITDMGMBRINDP1\_TEMP where RECIDXMBINP1 between :start\_id and :end\_id) b  ON (a.RECIDXMBINP1 = b.RECIDXMBINP1)  WHEN MATCHED THEN UPDATE SET  A.CRDTCARD=B.CRDTCARD,  A.BNKACCKEY01=B.BNKACCKEY01 ';\*/  /\* l\_sql\_stmt := 'EXEC vm1dta.DM2\_BQ9SC\_MB01\_MBRIND\_FORALL\_P(i\_scheduleName ,  i\_scheduleNumber,  i\_zprvaldYN,  i\_company,  i\_fsucocompany,  i\_usrprf,  i\_branch ,  i\_transCode,  i\_vrcmtime,  i\_vrcmuser,  i\_trdt1,  i\_acctYear,  i\_acctMonth,  i\_vrcmTermid,  :start\_id,  :end\_id)'; \*/  --l\_sql\_stmt:= 'EXEC vm1dta.DM2\_BQ9SC\_MB01\_MBRIND\_FORALL\_P(''FORPMBRP'', ''46'', ''N'', ''1'', ''9'', ''FORPMBRP'', ''31'', NULL, 231222, 180422, 0, 2018, 04, ''QPAD'', start\_id, end\_id)';  /\*l\_sql\_stmt1 := 'BEGIN vm1dta.DM2\_BQ9SC\_MB01\_MBRIND\_FORALL\_I('  ||''''||i\_scheduleName||''''||''''||','|| ''''||i\_scheduleNumber||'''', ''''||i\_zprvaldYN||'''', ''''||  i\_company||'''', ''''||i\_fsucocompany||'''', ''''||i\_usrprf||'''', ''''||i\_branch  || '''', ''''|| i\_transCode || '''', '''' ||i\_vrcmtime|| '''', ''''|| i\_vrcmuser|| '''', ''''|| i\_trdt1|| '''', ''''|| i\_acctYear|| '''', ''''||  i\_acctMonth|| '''', ''''|| i\_vrcmTermid|| ''''||','|| ''''||':start\_id, :end\_id);END;';\*/  dbms\_output.put\_line('l\_sql\_stmt1 :' || l\_sql\_stmt1);  l\_sql\_stmt := 'BEGIN vm1dta.DM2\_BQ9SC\_MB01\_MBRIND\_FORALL\_I(''FORPMBRP'', ''69'', ''N'', ''1'', ''9'', ''FORPMBRP'', ''31'', NULL, 231222, 180422, 0, 2018, 04, ''QPAD'', :start\_id, :end\_id);END;'  ;  dbms\_output.put\_line('l\_sql\_stmt :' || l\_sql\_stmt);  l\_sql\_stmt := 'BEGIN vm1dta.DM2\_BQ9SC\_MB01\_MBRIND\_FORALL\_I(''FORPMBRP'', ''69'', ''N'', ''1'', ''9'', ''FORPMBRP'', ''31'', NULL, 231222, 180422, 0, 2018, 04, ''QPAD'', :start\_id, :end\_id);END;'  ;  dbms\_parallel\_execute.run\_task(task\_name => 'DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK', sql\_stmt => l\_sql\_stmt, language\_flag => dbms\_sql  .native, parallel\_level => 24); ---Keep this number low may be 2 or max 5.  l\_status := dbms\_parallel\_execute.task\_status('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK');  dbms\_output.put\_line('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK completed! Status:' || l\_status);  --DBMS\_PARALLEL\_EXECUTE.DROP\_TASK('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK');  INSERT INTO vm1dta.dmbmonpf (  batch\_name,  start\_time,  end\_time,  job\_num,  datime,  remarks  ) VALUES (  'FORPMBRP',  v\_timestart,  dbms\_utility.get\_time,  69,  SYSDATE,  ' Loop 2\_1 remove with Append val'  );  COMMIT;  END IF;  END; |

1. Once you done please execute the proc from parallel module and see the magic.

Please refer below module:

|  |
| --- |
| CREATE OR REPLACE PROCEDURE "DM2\_BQ9SC\_MB01\_MBRIND\_FORALL\_I" (  i\_schedulename IN VARCHAR2,  i\_schedulenumber IN VARCHAR2,  i\_zprvaldyn IN VARCHAR2,  i\_company IN VARCHAR2,  i\_fsucocompany IN VARCHAR2,  i\_usrprf IN VARCHAR2,  i\_branch IN VARCHAR2,  i\_transcode IN VARCHAR2,  i\_vrcmtime IN NUMBER,  i\_vrcmuser IN NUMBER,  i\_trdt1 IN NUMBER,  i\_acctyear IN NUMBER,  i\_acctmonth IN NUMBER,  i\_vrcmtermid IN VARCHAR2,  start\_id IN NUMBER,  end\_id IN NUMBER  ) IS  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Amenment History: DMMB-01  \* Date Init Tag Decription  \* ----- ----- --- ---------------------------------------------------------------------------  \* MMMDD XXX MB1 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  \* 2804 001 MB2 included checking for inputfile 2 for ploicy not migrated  \* 0501 002 MB3 set values for ZCOLMCLS and ZPLANCLS, read t-table T3684 and get bnkacctyp  Change logic to set value of GCHIPF.BTDATENR  \* 0502 003 MB4 Change CRDATE in GCHIPF and set ZPENDDT in GCHPPF  \* 0505 RC MB5 #7865 - Incorrect fields on ZTIERPF  \* 0507 006 MB6 ITR4 - LOT2 changes  \* 0507 006 MB6 ITR4 - LOT2 changes  \* 0511 JDB MB7 Changes for data verification  \* 0512 MPS MB8 Changes for data verification  \* 0516 SJ MB9 Removed validation for TREFNUM and ZCONVINDPOL (now both columns can have value)  \* 0521 JDB MB10 Add EFFDATE for ztierpf  \* 0612 MPS MB11 TRANNO = TRANLUSED  \* 0625 JDB MB12 ZIRA ZJNDM-48  \* 0705 SC MB13 INCORRECT TRANNO FOR MEMBER FILE 2(Z-TRACKER 9564)  \* 0706 SC MB14 BILLFREQ AND GADJFREQ INCORRECTLY SET AS 00 (Z-TRACKER 9610)  \* AND COLUMN NAME "TRANSFERFLG" CHANGED TO "ZPDATATXFLG"  \* 0711 AK MB15 1) When STATCODE value is changed, corressponding value of DTETRM should also set  \* accordingly (except decline cases - ZTRXSTAT = 'RJ').  \* - If Policy is INFORCE, DTETRM should be max-date.  \* - If Policy is Cancelled, DTETRM should remain unchanged for value passed from TITDMGMBRINDP1.  \* 2) ZPOLDATE will remains unchanged, for value passed from TITDMGMBRINDP1.  \* 0729 MPS MB16 Removed chages for MB12 to get statcode  \* 1009 MPS MB17 Check of Credit card for Free plan  \*April\_18 JDB MB18 Performance optimization for Phase-2  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\*\*  Name Null? Type  ------------ -------- -----------------  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  v\_temp\_crdate NUMBER(8) DEFAULT 0;  /\*\*\*\* MB4 : MOD : condition change due to new requirement : END \*\*\*\*/  i\_text dmlog.LTEXT%TYPE;  --timecheck  v\_timestart NUMBER := dbms\_utility.get\_time;  ---local Vairables  v\_refnump1 VARCHAR2(8);  v\_refnumpseq titdmgmbrindp1.refnum@DATABASE\_LINK1%TYPE;  v\_seqno1 VARCHAR2(3);  seqmbrtmp1 NUMBER(18) DEFAULT 0;  seqmbrtmp2 NUMBER(18) DEFAULT 0;  v\_seqno VARCHAR2(1);  v\_seqnoincr VARCHAR2(3);  v\_isduplicatezdrppf NUMBER(1) DEFAULT 0;  v\_ismasterplmig NUMBER(1) DEFAULT 0;  v\_isduplicatep2 NUMBER(1) DEFAULT 0;  --v\_iszplancde NUMBER(1) DEFAULT 0;  v\_iszendcde NUMBER(1) DEFAULT 0;  v\_iszcmpcode NUMBER(1) DEFAULT 0;  v\_errorcountp1 NUMBER(1) DEFAULT 0;  v\_errorcountp2 NUMBER(1) DEFAULT 0;  v\_isanyerrorp1 VARCHAR2(1) DEFAULT 'N';  v\_isanyerrorp2 VARCHAR2(1) DEFAULT 'N';  -- v\_zdoecrtcountMB NUMBER(1) DEFAULT 0;  -- v\_zdoecrtcountIN NUMBER(1) DEFAULT 0;  v\_globalerrorp2 VARCHAR2(1) DEFAULT 'N';  v\_prefix VARCHAR2(2);  v\_refkeyp1 VARCHAR2(29 CHAR);  v\_refkeyp2 VARCHAR2(33 CHAR);  v\_isdtetrmvalid VARCHAR2(20 CHAR);  v\_iseffdate VARCHAR2(20 CHAR);  v\_timech01 VARCHAR2(10 CHAR);  v\_timech02 VARCHAR2(10 CHAR);  v\_zigvalue zdclpf.zigvalue%TYPE;  v\_template dfpopf.template%TYPE;  v\_zcovcmdt zesdpf.zcovcmdt%TYPE;  obj\_dfpopf dfpopf%rowtype;  v\_currency VARCHAR2(3 CHAR);  v\_zagptid zcpnpf.zagptid%TYPE;  v\_admnoper01 zagppf.admnoper05%TYPE;  v\_gagntsel01 zagppf.gagntsel05%TYPE;  v\_admnoper02 zagppf.admnoper05%TYPE;  v\_gagntsel02 zagppf.gagntsel05%TYPE;  v\_admnoper03 zagppf.admnoper05%TYPE;  v\_gagntsel03 zagppf.gagntsel05%TYPE;  v\_admnoper04 zagppf.admnoper05%TYPE;  v\_gagntsel04 zagppf.gagntsel05%TYPE;  v\_admnoper05 zagppf.admnoper05%TYPE;  v\_gagntsel05 zagppf.gagntsel05%TYPE;  --v\_zinstype ZSLPPF.zinstype%type;  v\_cltdob clntpf.cltdob%TYPE;  v\_zenspcd02\_1 zclepf.zenspcd02%TYPE;  v\_zcifcode zclepf.zcifcode%TYPE;  v\_bankaccdsc clbapf.bankaccdsc%TYPE;  v\_bnkactyp clbapf.bnkactyp%TYPE;  v\_bankkey clbapf.bankkey%TYPE;  v\_mthto clbapf.mthto%TYPE := 0;  v\_yearto clbapf.yearto%TYPE := 0;  v\_temp\_crdtcard titdmgmbrindp1.crdtcard@database\_link1%TYPE;  v\_trmdate titdmgmbrindp1.zpoltdate@database\_link1%TYPE; -- MB16  v\_zcrdtype zenctpf.zcrdtype%TYPE;  v\_polanv gchppf.polanv%TYPE;  v\_zagptnum gchipf.zagptnum%TYPE;  v\_zplancls gchppf.zplancls%TYPE;  v\_zcolmcls gchppf.zcolmcls%TYPE;  v\_zprmsi gmhipf.zprmsi%TYPE;  v\_olddta1 gmovpf.olddta%TYPE;  v\_newdta1 gmovpf.olddta%TYPE;  v\_olddta2 gmovpf.olddta%TYPE;  v\_newdta2 gmovpf.olddta%TYPE;  v\_olddta3 gmovpf.olddta%TYPE;  v\_newdta3 gmovpf.olddta%TYPE;  v\_refkey gmovpf.refkey%TYPE;  i\_trdt NUMBER(6) DEFAULT 0;  v\_isdatevalid VARCHAR2(20 CHAR);  v\_zpolcls zcpnpf.zpolcls%TYPE;  v\_zplancls\_new gchppf.zplancls%TYPE;  --v\_constructrefkey varchar2(4000 char);  v\_btdatenr gchipf.btdatenr%TYPE;  v\_datconage gmhdpf.age%TYPE;  --v\_temprefKeyp2 VARCHAR2(33 CHAR);  --v\_obj\_clba clbapf%rowtype;  v\_zclntid zendrpf.zclntid%TYPE;  v\_zfacthus zendrpf.zfacthus%TYPE; -- MB3  ----- WAVE 2---------  n\_issdate gmhipf.notsfrom%TYPE;  -----------Unique numbers------------  v\_seq\_gchppf gchppf.unique\_number%TYPE;  v\_seq\_gmhdpf gmhdpf.unique\_number%TYPE;  v\_seq\_gmhipf gmhipf.unique\_number%TYPE;  v\_seq\_gxhipf gxhipf.unique\_number%TYPE;  ---MB18  v\_seq\_gaphpf gaphpf.unique\_number%TYPE;  v\_seq\_gmovpf gmovpf.unique\_number%TYPE;  v\_seq\_zclepf zclepf.unique\_number%TYPE;  v\_seq\_chdrpf chdrpf.unique\_number%TYPE;  v\_seq\_gchipf gchipf.unique\_number%TYPE;  v\_seq\_gpsupf gpsupf.unique\_number%TYPE;  ---MB18  -----------Unique numbers------------  ----IG tables records---  obj\_gchd gchd%rowtype;  obj\_gchppf gchppf%rowtype;  obj\_gchipf gchipf%rowtype;  obj\_gmhdpf gmhdpf%rowtype;  obj\_gmhipf gmhipf%rowtype;  --obj\_zmcipf ZMCIPF%rowtype;  obj\_gxhipf gxhipf%rowtype;  obj\_gaphpf gaphpf%rowtype;  --obj\_mtrnpfPN MTRNPF%rowtype;  --obj\_mtrnpfCA MTRNPF%rowtype;  obj\_gmovpf1 gmovpf%rowtype;  obj\_gmovpf2 gmovpf%rowtype;  obj\_gmovpf3pn gmovpf%rowtype;  --obj\_gmovpf3CA GMOVPF%rowtype;  --obj\_ztierpf ZTIERPF%rowtype; --ITR3  obj\_zclepf zclepf%rowtype; --ITR3  obj\_ztierpf view\_dm\_ztierpf%rowtype; --ITR3  obj\_ztempcovpf1 view\_dm\_ztempcovpf%rowtype; --SIT  obj\_ztempcovpf2 view\_dm\_ztempcovpf%rowtype; --SIT  obj\_ztemptierpf1 view\_dm\_ztemptierpf%rowtype; --SIT  obj\_ztemptierpf2 view\_dm\_ztemptierpf%rowtype; --SIT  obj\_zcelinkpf view\_dm\_zcelinkpf%rowtype; --ITR4 CHNAGES  obj\_gpsupf gpsupf%rowtype;  obj\_getmpol ig\_dm\_masterpol%rowtype;  obj\_getclnt clntpf%rowtype;  ------Constant---------  /\*---------RH-CHANGES---------  TYPE obj\_mbrmig IS RECORD(  refnum TITDMGMBRINDp1.refnum@DATABASE\_LINK1%type,  mpolnum TITDMGMBRINDp1.mpolnum@DATABASE\_LINK1%type,  statcode TITDMGMBRINDp1.statcode@DATABASE\_LINK1%type,  dtetrm TITDMGMBRINDp1.dtetrm@DATABASE\_LINK1%type,  zplancde TITDMGMBRINDp1.zplancde@DATABASE\_LINK1%type,  ZWAITPEDT TITDMGMBRINDp1.ZWAITPEDT@DATABASE\_LINK1%type,  effdate TITDMGMBRINDp1.effdate@DATABASE\_LINK1%type);  TYPE v\_array IS TABLE OF obj\_mbrmig;  mbrmigindex integer := 0;  mbrmig\_list v\_array;  ------------RH\_CHANGES------------ \*/  i\_refnum titdmgmbrindp1.refnum@database\_link1%TYPE;  i\_cnttypind titdmgmbrindp1.cnttypind@database\_link1%TYPE;  i\_mpolnum titdmgmbrindp1.mpolnum@database\_link1%TYPE;  i\_statcode titdmgmbrindp1.statcode@database\_link1%TYPE;  i\_dtetrm titdmgmbrindp1.dtetrm@database\_link1%TYPE;  i\_zplancde titdmgmbrindp1.zplancde@database\_link1%TYPE;  i\_zwaitpedt titdmgmbrindp1.zwaitpedt@database\_link1%TYPE;  i\_effdate titdmgmbrindp1.effdate@database\_link1%TYPE;  i\_zpoltdate titdmgmbrindp1.zpoltdate@database\_link1%TYPE; --MB16  i\_zpdatatxflag titdmgmbrindp1.zpdatatxflg@database\_link1%TYPE; --MB16  i\_ztrxstat titdmgmbrindp1.ztrxstat@database\_link1%TYPE; --MB16  i\_trmflag VARCHAR2(1 CHAR);  ------------RH\_CHANGES------------  c\_prefix\_mebr CONSTANT VARCHAR2(2) := get\_migration\_prefix('MEBR', i\_company); --'MB'  c\_prefix\_indv CONSTANT VARCHAR2(2) := get\_migration\_prefix('INDV', i\_company); --'IN'  /\* C\_PREFIX\_MEBR constant varchar2(2) := 'MB';  C\_PREFIX\_INDV constant varchar2(2) := 'IN';\*/  c\_space CONSTANT VARCHAR2(1) := ' ';  c\_zero CONSTANT NUMBER := 0;  c\_maxdate CONSTANT VARCHAR2(20 CHAR) := '99999999';  c\_errorcountp1 CONSTANT NUMBER := 5;  c\_errorcountp2 CONSTANT NUMBER := 5;  c\_z001 CONSTANT VARCHAR2(4) := 'RQLH';  /\* RQLH Z001 Policy already migrated \*/  c\_z002 CONSTANT VARCHAR2(4) := 'RQLI';  /\* RQLI Z002 Client not yet migrated \*/  c\_z003 CONSTANT VARCHAR2(4) := 'RQLJ';  /\* RQLJ Z003 Master policy not yet migrated \*/  c\_z004 CONSTANT VARCHAR2(4) := 'RQLK';  /\* RQLK Z004 Input file 2 records does not exist \*/  c\_z005 CONSTANT VARCHAR2(4) := 'RQLL';  /\* RQLL Z005 Sales Channel not in TQ9FW \*/  c\_z006 CONSTANT VARCHAR2(4) := 'RQLM';  /\* RQLM Z006 Relation not in T3584 \*/  c\_z007 CONSTANT VARCHAR2(4) := 'RQLN';  /\* RQLN Z007 Transaction Status not in TQ9FT \*/  c\_z008 CONSTANT VARCHAR2(4) := 'RQLO';  /\* RQLO Z008 Status Reason not in TQ9FU \*/  c\_z009 CONSTANT VARCHAR2(4) := 'RQLP';  /\* RQLP Z009 Sales Plan not valid \*/  c\_z010 CONSTANT VARCHAR2(4) := 'RQLQ';  /\* RQLQ Z010 Endorser Code Not valid \*/  c\_z011 CONSTANT VARCHAR2(4) := 'RQQ1';  /\* RQQ1 Z011 Credit Card No/Bank Account No/ Endorser specific code is blank \*/  c\_z012 CONSTANT VARCHAR2(4) := 'RQLS';  /\* RQLS Z012 Invalid contract type indicator \*/  c\_z013 CONSTANT VARCHAR2(4) := 'RQLT';  /\* RQLT Z013 Invalid Date \*/  c\_z014 CONSTANT VARCHAR2(4) := 'RQLU';  /\* RQLU Z014 Product code not in T9797 \*/  c\_z021 CONSTANT VARCHAR2(4) := 'RQM1';  /\* RQM1 Z021 Campaign Code not valid \*/  c\_z099 CONSTANT VARCHAR2(4) := 'RQO6';  /\* RQO6 Z099 Duplicate record found \*/  --C\_Z075 constant varchar2(4) := 'RQNJ'; /\* RQNJ Z075 Credit card is mandatory \*/  --C\_Z076 constant varchar2(4) := 'RQNK'; /\* RQNK Z076 PREAUTNO is mandatory \*/  --C\_Z077 constant varchar2(4) := 'RQNL'; /\* RQNL Z077 BANKACCNO is mandatory \*/  --C\_Z078 constant varchar2(4) := 'RQNM'; /\* RQNM Z078 ZENDCDE is mandatory \*/  --C\_Z079 constant varchar2(4) := 'RQNN'; /\* RQNN Z079 Invalid ZENSPCD02 \*/  --C\_Z080 constant varchar2(4) := 'RQNO'; /\* RQNO Z080 ZENSPCD not in ZCLEPF \*/  --C\_Z081 constant varchar2(4) := 'RQNP'; /\* RQNP Z081 GPOLTYPE not set - DFPOPF \*/  c\_z082 CONSTANT VARCHAR2(4) := 'RQNQ';  /\* RQNQ Z082 REFNUM is mandatory \*/  c\_z083 CONSTANT VARCHAR2(4) := 'RQNR';  /\* RQNR Z083 GPOLTYPE is mandatory \*/  c\_z084 CONSTANT VARCHAR2(4) := 'RQNS';  /\* RQNS Z084 Master Pol. is required \*/  c\_z085 CONSTANT VARCHAR2(4) := 'RQNT';  /\* RQNT Z085 Master Pol. not required \*/  c\_z086 CONSTANT VARCHAR2(4) := 'RQNU';  /\* RQNU Z086 ZPOLPERD is mandatory \*/  c\_z087 CONSTANT VARCHAR2(4) := 'RQNV';  /\* RQNV Z087 CLTRELN is mandatory \*/  c\_z088 CONSTANT VARCHAR2(4) := 'RQNW';  /\* RQNW Z088 ZPLANCDE is mandatory \*/  c\_z089 CONSTANT VARCHAR2(4) := 'RQNX';  /\* RQNX Z089 DTERM not required \*/  c\_z090 CONSTANT VARCHAR2(4) := 'RQNY';  /\* RQNY Z090 ZCMPCODE is mandatory \*/  c\_z091 CONSTANT VARCHAR2(4) := 'RQNZ';  /\* RQNZ Z091 PRODTYP is mandatory \*/  c\_z092 CONSTANT VARCHAR2(4) := 'RQO0';  /\* RQO0 Z092 HSUMINSU is mandatory \*/  c\_e315 CONSTANT VARCHAR2(4) := 'E315';  c\_f623 CONSTANT VARCHAR2(4) := 'F623';  c\_s008 CONSTANT VARCHAR2(4) := 'S008';  --C\_RPRD constant varchar2(4) := 'RPRD';  --C\_F950 constant varchar2(4) := 'F950';  --C\_RPRM constant varchar2(4) := 'RPRM';  --C\_RFTQ constant varchar2(4) := 'RFTQ';  --C\_F826 constant varchar2(4) := 'F826';  --C\_W533 constant varchar2(4) := 'W533';  c\_bq9sc CONSTANT VARCHAR2(5) := 'BQ9SC';  --C\_RPRK constant varchar2(4) := 'RPRK';  --C\_RPRF constant varchar2(4) := 'RPRF';  ----ITR3 Validation error codes:START  c\_z093 CONSTANT VARCHAR2(4) := 'RQO1';  /\* ZCONVINDPOL must be blank\*/  c\_z120 CONSTANT VARCHAR2(4) := 'RQOQ';  /\* ZCONVINDPOL is invalid\*/  c\_z095 CONSTANT VARCHAR2(4) := 'RQO3';  /\* Old Pol No must be blank \*/  c\_z094 CONSTANT VARCHAR2(4) := 'RQO2';  /\* Invalid Old Policy No.\*/  c\_z121 CONSTANT VARCHAR2(4) := 'RQOR';  /\* TREFNUM is invalid \*/  c\_z096 CONSTANT VARCHAR2(4) := 'RQO4';  /\* Outside policy period\*/  c\_z097 CONSTANT VARCHAR2(4) := 'RQO5';  /\* From date > To date\*/  /\*\*\*\* ITR-4\_LOT2 : MB6 - MOD : condition change due to new requirement : START \*\*\*\*/  c\_z028 CONSTANT VARCHAR2(4) := 'E315';  /\* Must be Y or N\*/  /\*\*\*\* ITR-4\_LOT2 : MB6 - MOD : condition change due to new requirement : END \*\*\*\*/  ----ITR3 Validation error codes:END  c\_trmpol CONSTANT VARCHAR2(2) := '13';  c\_addmbr CONSTANT VARCHAR2(2) := '01';  c\_newbz\_issue CONSTANT VARCHAR2(4) := 'T903';  c\_iss\_term CONSTANT VARCHAR2(4) := 'T913';  c\_bigdecimal\_default CONSTANT NUMBER := 0.00;  c\_bigdecimal\_default1 CONSTANT NUMBER := 0.000;  c\_recordskipped CONSTANT VARCHAR2(17) := 'Record skipped';  c\_recordsuccess CONSTANT VARCHAR2(20) := 'Record successful';  c\_success CONSTANT VARCHAR2(3) := 'S';  c\_error CONSTANT VARCHAR2(3) := 'E';  -- C\_RECSKIPPEDUPDATE constant varchar2(17) := '''Record skipped''';  --C\_SUCCESS constant varchar2(3) := '''S''';  -- C\_ERROR constant varchar2(3) := '''E''';  c\_funcdeaddmbr CONSTANT VARCHAR2(5) := '04000';  c\_funcdeaddmbrpp CONSTANT VARCHAR2(5) := '05000';  c\_funcdetrmbr CONSTANT VARCHAR2(5) := '04021';  c\_funcdetrmbrpp CONSTANT VARCHAR2(5) := '05021';  --TITDMGMBRINDP1  ----MB18:START---------  CURSOR cur\_mbr\_ind\_p1 IS  /\* SELECT \*  FROM TITDMGMBRINDP1@DATABASE\_LINK1  ORDER BY LPAD(refnum, 15, '0') ASC;\*/  SELECT  a.\*,  c.cltdob  FROM  vm1dta.titdmgmbrindp1\_p a  INNER JOIN vm1dta.zdclpf\_1412 b ON substr(a.refnum, 1, 8) = b.zentity  LEFT OUTER JOIN vm1dta.clntpf c ON b.zigvalue = c.clntnum  WHERE  a.recidxmbinp1 BETWEEN start\_id AND end\_id  ORDER BY  lpad(a.refnum, 15, '0') ASC;--freeplan  /\* SELECT \*  FROM TITDMGMBRINDP1@DATABASE\_LINK1  where substr(refnum, 1, 8) IN ('00287024','00287571','002875800');\*/  TYPE t\_mbrind1\_list IS  TABLE OF cur\_mbr\_ind\_p1%rowtype;  mbrind1\_list t\_mbrind1\_list;  ----MB18:END---------  --where TRIM(refnum) = '03671288000';  /\* select \*  from (select \*  from TITDMGMBRINDP1@DATABASE\_LINK1  order by TO\_NUMBER(refnum) asc)  where rownum < 10; \*/  obj\_mbrindp1 cur\_mbr\_ind\_p1%rowtype;  --TITDMGMBRINDP2  /\*CURSOR cur\_mbr\_ind\_p2(p1refnum TITDMGMBRINDP1.REFNUM@DATABASE\_LINK1%type) IS  SELECT \*  FROM TITDMGMBRINDP2@DATABASE\_LINK1  where TRIM(refnum) = p1refnum;\*/  /\*CURSOR cur\_mbr\_ind\_p2 IS  SELECT \*  FROM TITDMGMBRINDP2@DATABASE\_LINK1  order by LPAD(refnum, 15, '0') asc; \*/  ----MB18:START---------  CURSOR cur\_mbr\_ind\_p2 IS  SELECT  a.\*,  b.chdrnum AS chdrnum\_b,  c.refnum AS refnum\_c,  c.cnttypind AS cnttypind\_c,  c.mpolnum AS mpolnum\_c,  c.statcode AS statcode\_c,  c.dtetrm AS dtetrm\_c,  c.zplancde AS zplancde\_c,  c.zwaitpedt AS zwaitpedt\_c,  c.effdate AS effdate\_c,  c.zpoltdate AS zpoltdate\_c,  c.zpdatatxflg AS zpdatatxflg\_c,  c.ztrxstat AS ztrxstat\_c  FROM  vm1dta.titdmgmbrindp2\_p a  LEFT OUTER JOIN vm1dta.zdrppf b ON substr(a.refnum, 1, 8) = b.chdrnum  INNER JOIN vm1dta.titdmgmbrindp1\_p c ON a.refnum = c.refnum  WHERE  a.recidxmbindp3 BETWEEN start\_id AND end\_id  ORDER BY  lpad(a.refnum, 15, '0') ASC;  --Removed trim from chdrnum  /\*SELECT A.\*, B.CHDRNUM AS CHDRNUM\_B  FROM TITDMGMBRINDP2@DATABASE\_LINK1 A, cur\_mbr\_ind\_p1 C  LEFT OUTER JOIN VM1DTA.ZDRPPF B  ON SUBSTR(A.refnum, 1, 8) = B.CHDRNUM  WHERE A.refnum between start\_id and end\_id  ORDER BY LPAD(A.refnum, 15, '0') ASC;\*/  /\*select \* from (SELECT A.\*, B.CHDRNUM AS CHDRNUM\_B  FROM TITDMGMBRINDP2@DATABASE\_LINK1 A  LEFT OUTER JOIN VM1DTA.ZDRPPF B  ON SUBSTR(A.refnum, 1, 8) = TRIM(B.CHDRNUM))  where SUBSTR(refnum, 1, 8)IN ('00287024','00287571','002875800');\*/  TYPE t\_mbrind2\_list IS  TABLE OF cur\_mbr\_ind\_p2%rowtype;  mbrind2\_list t\_mbrind2\_list;  ----MB18:END---------  -- where TRIM(refnum) = p1refnum;  --obj\_mbrindp2 TITDMGMBRINDP2@DATABASE\_LINK1%rowtype;  obj\_mbrindp2 cur\_mbr\_ind\_p2%rowtype;  --------------Common Function Start---------  --v\_tablecnt NUMBER(1) := 0;  v\_tablenametemp VARCHAR2(10);  v\_tablenamemb VARCHAR2(10);  v\_tablenamein VARCHAR2(10);  o\_defaultvalues pkg\_dm\_common\_operations.defaultvaluesmap;  --itemexist pkg\_dm\_common\_operations.itemschec;  itemexist pkg\_common\_dmmb.itemschec;  o\_errortext pkg\_dm\_common\_operations.errordesc;  i\_zdoe\_infop1 pkg\_dm\_common\_operations.obj\_zdoe;  i\_zdoe\_infop2 pkg\_dm\_common\_operations.obj\_zdoe;  i\_zdoe\_err\_infop2 pkg\_dm\_common\_operations.obj\_zdoe;  -- checkrefnum pkg\_common\_dmmb.refnumtype;  getzigvalue pkg\_common\_dmmb.zigvaluetype;  checksalplan pkg\_common\_dmmb.salplantype;  getzinstype pkg\_common\_dmmb.zinstype;  getdfpo pkg\_common\_dmmb.dfpopftype;  getclba pkg\_common\_dmmb.clbatype;  checkchdrnum pkg\_common\_dmmb.gchdtype;  zendcde pkg\_common\_dmmb.checkzendcde;  campcode pkg\_common\_dmmb.checkcampcode;  getmpol pkg\_common\_dmmb.mpoltype;  getclntinfo pkg\_common\_dmmb.clntpftype;  checkpoldup pkg\_common\_dmmb.polduplicatetype;  clntdob pkg\_common\_dmmb.getclntdob;  facthouse pkg\_common\_dmmb.zendfacthouse; -- MB3  -----------RH:Changes-----------  mbrp1info pkg\_common\_dmmb.mbrinfotype;  -----------RH:Changes-----------  -- getpolanv pkg\_common\_dmmb.plolanvtype;  TYPE ercode\_tab IS  TABLE OF VARCHAR(4) INDEX BY BINARY\_INTEGER;  t\_ercodep1 ercode\_tab;  t\_ercodep2 ercode\_tab;  TYPE errorfield\_tab IS  TABLE OF VARCHAR(10) INDEX BY BINARY\_INTEGER;  t\_errorfieldp1 errorfield\_tab;  t\_errorfieldp2 errorfield\_tab;  TYPE errormsg\_tab IS  TABLE OF VARCHAR(250) INDEX BY BINARY\_INTEGER;  t\_errormsgp1 errormsg\_tab;  t\_errormsgp2 errormsg\_tab;  TYPE errorfieldvalue\_tab IS  TABLE OF VARCHAR(2000) INDEX BY BINARY\_INTEGER;  t\_errorfieldvalp1 errorfieldvalue\_tab;  t\_errorfieldvalp2 errorfieldvalue\_tab;  TYPE errorprofram\_tab IS  TABLE OF VARCHAR(10) INDEX BY BINARY\_INTEGER;  t\_errorprogramp1 errorprofram\_tab;  t\_errorprogramp2 errorprofram\_tab;  TYPE obj\_zdoe IS RECORD (  i\_tablecnt NUMBER(1),  i\_tablename VARCHAR2(10),  i\_refkey zdoepf.zrefkey%TYPE,  i\_zfilename zdoepf.zfilenme%TYPE,  i\_indic zdoepf.indic%TYPE,  i\_prefix VARCHAR2(2),  i\_scheduleno zdoepf.jobnum%TYPE,  i\_error01 zdoepf.eror01%TYPE,  i\_errormsg01 zdoepf.errmess01%TYPE,  i\_errorfield01 zdoepf.erorfld01%TYPE,  i\_fieldvalue01 zdoepf.fldvalu01%TYPE,  i\_errorprogram01 zdoepf.erorprog01%TYPE  );  obj\_error obj\_zdoe;  ---------------Common function end-----------  --v\_faltu VARCHAR2(1) DEFAULT 'N';  TYPE error\_type IS  TABLE OF obj\_zdoe;  error\_list error\_type := error\_type();  errindex INTEGER := 0;  ----------------  v\_insendate gchi.insendte%TYPE; --MB12  v\_busdate busdpf.busdate%TYPE; --MB12  --MB16 TYPE checkstat IS TABLE OF VARCHAR2(50) INDEX BY VARCHAR2(50); --MB12  --MB16 a\_checkpolst checkstat;  --MB16 b\_checkpolst checkstat;  ------MB18-----------------------  -------FORALL-----  idx PLS\_INTEGER;  TYPE gchd\_type IS  TABLE OF gchd%rowtype;  gchd\_list gchd\_type := gchd\_type();  gchdindex INTEGER := 0;  TYPE gchppf\_type IS  TABLE OF gchppf%rowtype;  gchppf\_list gchppf\_type := gchppf\_type();  gchppfindex INTEGER := 0;  TYPE gchipf\_type IS  TABLE OF gchipf%rowtype;  gchipf\_list gchipf\_type := gchipf\_type();  gchipfindex INTEGER := 0;  TYPE gmhdpf\_type IS  TABLE OF gmhdpf%rowtype;  gmhdpf\_list gmhdpf\_type := gmhdpf\_type();  gmhdpfindex INTEGER := 0;  TYPE gmovpf1\_type IS  TABLE OF gmovpf%rowtype;  gmovpf1\_list gmovpf1\_type := gmovpf1\_type();  gmovpf1index INTEGER := 0;  TYPE gmhipf\_type IS  TABLE OF gmhipf%rowtype;  gmhipf\_list gmhipf\_type := gmhipf\_type();  gmhipfindex INTEGER := 0;  TYPE gmovpf2\_type IS  TABLE OF gmovpf%rowtype;  gmovpf2\_list gmovpf2\_type := gmovpf2\_type();  gmovpf2index INTEGER := 0;  TYPE zclepf\_type IS  TABLE OF zclepf%rowtype;  zclepf\_list zclepf\_type := zclepf\_type();  zclepfindex INTEGER := 0;  TYPE gpsupf\_type IS  TABLE OF gpsupf%rowtype;  gpsupf\_list gpsupf\_type := gpsupf\_type();  gpsupfindex INTEGER := 0;  TYPE zcelinkpf\_type IS  TABLE OF view\_dm\_zcelinkpf%rowtype;  zcelinkpf\_list zcelinkpf\_type := zcelinkpf\_type();  zcelinkpfindex INTEGER := 0;  ----------Loop2  TYPE gxhipf\_type IS  TABLE OF gxhipf%rowtype;  gxhipf\_list gxhipf\_type := gxhipf\_type();  gxhipfindex INTEGER := 0;  TYPE gmovpf3pn\_type IS  TABLE OF gmovpf%rowtype;  gmovpf3pn\_list gmovpf3pn\_type := gmovpf3pn\_type();  gmovpf3pnindex INTEGER := 0;  TYPE gaphpf\_type IS  TABLE OF gaphpf%rowtype;  gaphpf\_list gaphpf\_type := gaphpf\_type();  gaphpfindex INTEGER := 0;  TYPE ztierpf\_type IS  TABLE OF view\_dm\_ztierpf%rowtype;  ztierpf\_list ztierpf\_type := ztierpf\_type();  ztierpfindex INTEGER := 0;  TYPE ztempcovpf1\_type IS  TABLE OF view\_dm\_ztempcovpf%rowtype;  ztempcovpf1\_list ztempcovpf1\_type := ztempcovpf1\_type();  ztempcovpf1index INTEGER := 0;  TYPE ztempcovpf2\_type IS  TABLE OF view\_dm\_ztempcovpf%rowtype;  ztempcovpf2\_list ztempcovpf2\_type := ztempcovpf2\_type();  ztempcovpf2index INTEGER := 0;  TYPE ztemptierpf1\_type IS  TABLE OF view\_dm\_ztemptierpf%rowtype;  ztemptierpf1\_list ztemptierpf1\_type := ztemptierpf1\_type();  ztemptierpf1index INTEGER := 0;  TYPE ztemptierpf2\_type IS  TABLE OF view\_dm\_ztemptierpf%rowtype;  ztemptierpf2\_list ztemptierpf2\_type := ztemptierpf2\_type();  ztemptierpf2index INTEGER := 0;  tablecount NUMBER := 0;  ----FORALL----------  ------MB18-----------------------  BEGIN  v\_tablenametemp := 'ZDOE'  || trim(c\_prefix\_mebr)  || lpad(trim(i\_schedulenumber), 4, '0');  v\_tablenamemb := trim(v\_tablenametemp);  v\_tablenametemp := 'ZDOE'  || trim(c\_prefix\_indv)  || lpad(trim(i\_schedulenumber), 4, '0');  v\_tablenamein := trim(v\_tablenametemp);  SELECT  COUNT(1)  INTO tablecount  FROM  all\_tables  WHERE  owner = 'VM1DTA'  AND table\_name = v\_tablenamemb;  --if(tablecount = 0)then  /\* DBMS\_PROFILER.start\_profiler('DM2\_MBRIND\_1' ||  TO\_CHAR(SYSDATE, 'YYYYMMDD HH24:MI:SS'));\*/  SELECT  busdate  INTO v\_busdate  FROM  busdpf  WHERE  TRIM(company) = TRIM(i\_company); --MB12 : BUSDATE  --------Master plocies loading----------  --MB18: START : Commenting below query becasue of parallel issue  /\*DELETE FROM IG\_DM\_MASTERPOL;  INSERT INTO IG\_DM\_MASTERPOL  SELECT DISTINCT (a.CHDRNUM),  b.ZPLANCLS,  b.Zcolmcls,  b.POLANV,  c.ZAGPTNUM  FROM gchd a, GCHPPF b, GCHIPF c  WHERE a.CHDRNUM = b.CHDRNUM  AND a.CHDRNUM = c.CHDRNUM  AND a.CHDRNUM = a.MPLNUM;\*/  ----------------------  --MB18: END : Commenting below query becasue of parallel issue  ---------Common Function Calling---------  pkg\_common\_dmmb.getmpolinfo(getmpol => getmpol);  pkg\_dm\_common\_operations.getdefaultvalues(i\_itemname => c\_bq9sc, i\_company => i\_company, o\_defaultvalues => o\_defaultvalues);  pkg\_dm\_common\_operations.geterrordesc(i\_module\_name => 'DMMB', o\_errortext => o\_errortext);  pkg\_common\_dmmb.getitemvalue(itemexist => itemexist);  pkg\_common\_dmmb.getzigvalue(getzigvalue => getzigvalue);  pkg\_common\_dmmb.checksalplan(checksalplan => checksalplan);  pkg\_common\_dmmb.getzinstype(getzinstype => getzinstype);  pkg\_common\_dmmb.getdfpo(getdfpo => getdfpo);  --MB 18: FOR PGA not used pkg\_common\_dmmb.getclba(getclba => getclba);  pkg\_common\_dmmb.checkendorser(zendcde => zendcde);  pkg\_common\_dmmb.checkcampcde(campcode => campcode);  pkg\_common\_dmmb.checkpoldup(checkpoldup => checkpoldup);  --MB 18: For PGA pkg\_common\_dmmb.checkclntdob(clntdob => clntdob);  --MB 18: For PGA pkg\_common\_dmmb.getmbrp1info(mbrp1info => mbrp1info);  pkg\_common\_dmmb.getfacthouse(facthouse => facthouse); -- MB3  ---------Common Function Calling------------  -- pkg\_dm\_common\_operations.createzdoepf(i\_tablename => v\_tableNameMB);  --pkg\_dm\_common\_operations.createzdoepf(i\_tablename => v\_tableNameIN);  --end if;  /\*SELECT COUNT(\*)  INTO v\_zdoecrtcountMB  FROM user\_tables  WHERE TRIM(TABLE\_NAME) = v\_tableNameMB;  SELECT COUNT(\*)  INTO v\_zdoecrtcountIN  FROM user\_tables  WHERE TRIM(TABLE\_NAME) = v\_tableNameIN;  \*/  i\_trdt := substr(i\_trdt1, 3, 6);  /\* OPEN cur\_mbr\_ind\_p1;  <<skipRecord>>  LOOP  FETCH cur\_mbr\_ind\_p1  INTO obj\_mbrindp1;  EXIT WHEN cur\_mbr\_ind\_p1%notfound;\*/  ---MB18:  OPEN cur\_mbr\_ind\_p1;  LOOP  FETCH cur\_mbr\_ind\_p1 BULK COLLECT INTO mbrind1\_list LIMIT 5000;  << skiprecord >>  ---MB18:  FOR i IN 1..mbrind1\_list.count LOOP  -- dbms\_output.put\_line('Loop 1 start');  i\_text := start\_id  || '-'  || end\_id;  dmlog\_info(i\_lkey => 'Parallel excution in loop', i\_ltext => i\_text);  obj\_mbrindp1 := mbrind1\_list(i);  IF trim(obj\_mbrindp1.cnttypind) = 'I' THEN  v\_prefix := c\_prefix\_indv;  ELSE  v\_prefix := c\_prefix\_mebr;  END IF;  v\_refkeyp1 := substr(trim(obj\_mbrindp1.refnum), 1, 8);  v\_refnump1 := substr(trim(obj\_mbrindp1.refnum), 1, 8);  v\_refnumpseq := trim(obj\_mbrindp1.refnum);  v\_seqno1 := substr(trim(obj\_mbrindp1.refnum), 9, 3);  v\_seqnoincr := v\_seqno1 + 1;  v\_seqno := 1;  i\_trmflag := 'N';  ----------Initialization-------  i\_zdoe\_infop1 := NULL;  i\_zdoe\_infop1.i\_zfilename := 'TITDMGMBRINDP1';  i\_zdoe\_infop1.i\_prefix := v\_prefix;  i\_zdoe\_infop1.i\_scheduleno := i\_schedulenumber;  i\_zdoe\_infop1.i\_refkey := trim(v\_refkeyp1);  IF ( trim(v\_prefix) = 'MB' ) THEN  i\_zdoe\_infop1.i\_tablename := v\_tablenamemb;  ELSE  i\_zdoe\_infop1.i\_tablename := v\_tablenamein;  END IF;  -- v\_tablecnt := 1;  v\_isanyerrorp1 := 'N';  v\_errorcountp1 := 0;  t\_ercodep1(1) := NULL;  t\_ercodep1(2) := NULL;  t\_ercodep1(3) := NULL;  t\_ercodep1(4) := NULL;  t\_ercodep1(5) := NULL;  errindex := 0;  v\_zprmsi := 0;  ----------Initialization -------  --dbms\_output.put\_line('TRIM(v\_refnump1)==>' || TRIM(v\_refnump1));  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  v\_temp\_crdate := 0;  SELECT  to\_number(TO\_CHAR(TO\_DATE(obj\_mbrindp1.crdate, 'yyyymmdd') - 1, 'yyyymmdd'))  INTO v\_temp\_crdate  FROM  dual;  -- dbms\_output.put\_line(obj\_mbrindp1.crdate || ' ' || v\_temp\_crdate );  /\*\*\*\* MB4 : MOD : condition change due to new requirement : END \*\*\*\*/  ---------------First part of validation -TITDMGMBRINDP1----------------------------------------  --- MB12 : MOD : Change for state code based on termination conditions : start  ---IF zpoltdate and dtetrm = MAX DATE  --- MB16 START --  IF ( trim(obj\_mbrindp1.ztrxstat) ) <> 'RJ' THEN  v\_trmdate := c\_maxdate;  IF ( trim(obj\_mbrindp1.zpoltdate) <> c\_maxdate ) AND ( TRIM(obj\_mbrindp1.zpoltdate) IS NOT NULL ) THEN  v\_trmdate := obj\_mbrindp1.zpoltdate;  END IF;  IF ( trim(obj\_mbrindp1.dtetrm) <> c\_maxdate ) AND ( TRIM(obj\_mbrindp1.dtetrm) IS NOT NULL ) THEN  v\_trmdate := obj\_mbrindp1.dtetrm;  END IF;  -- dbms\_output.put\_line('CHDRNUM ' || obj\_mbrindp1.refnum);  -- dbms\_output.put\_line('ZDTETRM ' || v\_trmdate);  IF ( v\_trmdate <> c\_maxdate ) AND ( trim(obj\_mbrindp1.zpdatatxflg) = 'Y' ) THEN  -- dbms\_output.put\_line('Mig Date ' || v\_BUSDATE);  IF ( v\_trmdate >= v\_busdate ) THEN  obj\_mbrindp1.statcode := 'IF';  obj\_mbrindp1.zpoltdate := v\_trmdate;  obj\_mbrindp1.dtetrm := c\_maxdate;  END IF;  IF ( v\_trmdate < v\_busdate ) THEN  obj\_mbrindp1.statcode := 'CA';  obj\_mbrindp1.zpoltdate := c\_maxdate;  obj\_mbrindp1.dtetrm := v\_trmdate;  END IF;  END IF;  -- dbms\_output.put\_line('OUT Date ' || v\_BUSDATE);  IF ( v\_trmdate <> c\_maxdate ) AND ( ( trim(obj\_mbrindp1.zpdatatxflg) <> 'Y' ) OR ( TRIM(obj\_mbrindp1.zpdatatxflg  ) IS NULL ) ) THEN  -- dbms\_output.put\_line('Mig Date 2' || v\_BUSDATE);  obj\_mbrindp1.statcode := 'IF';  obj\_mbrindp1.zpoltdate := v\_trmdate;  obj\_mbrindp1.dtetrm := c\_maxdate;  END IF;  END IF;  -- IF ((TRIM(obj\_mbrindp1.zpoltdate) = C\_MAXDATE) AND  -- (TRIM(obj\_mbrindp1.dtetrm) = C\_MAXDATE)) THEN  -- obj\_mbrindp1.STATCODE := obj\_mbrindp1.statcode;  -- END IF;  -- IF ((TRIM(obj\_mbrindp1.zpoltdate) <> C\_MAXDATE) OR  -- (TRIM(obj\_mbrindp1.dtetrm) <> C\_MAXDATE)) THEN  -- IF (TRIM(obj\_mbrindp1.ZTRXSTAT)) <> 'RJ' THEN --MB15 -- IF\_starts (STATCODE should not be reset for Decilned polices)  ----for zpoltdate != MAX Date  -- IF ((TRIM(obj\_mbrindp1.zpoltdate) IS NOT NULL) AND  -- (TRIM(obj\_mbrindp1.zpoltdate) != 0) AND  -- (TRIM(obj\_mbrindp1.zpoltdate) <> C\_MAXDATE)) THEN  ----Condition TRANSFERFLG(ZPDATATXFLG) = 'Y'  -- IF (TRIM(obj\_mbrindp1.ZPDATATXFLG) = 'Y') THEN ---MB14  -- IF (v\_BUSDATE <= obj\_mbrindp1.zpoltdate) THEN  -- obj\_mbrindp1.STATCODE := 'IF';  -- obj\_mbrindp1.dtetrm := C\_MAXDATE; -- MB15  -- END IF;  -- IF (obj\_mbrindp1.zpoltdate < v\_BUSDATE) THEN  -- obj\_mbrindp1.STATCODE := 'CA';  -- obj\_mbrindp1.dtetrm := obj\_mbrindp1.zpoltdate; --MB15  -- END IF;  -- ELSE  ----Condition TRANSFERFLG(ZPDATATXFLG) != 'Y'  -- obj\_mbrindp1.STATCODE := 'IF';  -- obj\_mbrindp1.dtetrm := C\_MAXDATE; -- MB15  -- END IF;  -- END IF;  ----for dtetrm != MAX Date  -- IF ((TRIM(obj\_mbrindp1.dtetrm) IS NOT NULL) AND  -- (TRIM(obj\_mbrindp1.dtetrm) != 0) AND  -- (TRIM(obj\_mbrindp1.dtetrm) <> C\_MAXDATE)) THEN  ----Condition TRANSFERFLG(ZPDATATXFLG) = 'Y'  -- IF (TRIM(obj\_mbrindp1.ZPDATATXFLG) = 'Y') THEN ---MB14  -- IF (v\_BUSDATE <= obj\_mbrindp1.dtetrm) THEN  -- obj\_mbrindp1.STATCODE := 'IF';  -- obj\_mbrindp1.zpoltdate := obj\_mbrindp1.dtetrm; -- MB15  -- obj\_mbrindp1.dtetrm := C\_MAXDATE; -- MB15  -- END IF;  -- IF (obj\_mbrindp1.dtetrm < v\_BUSDATE) THEN  -- obj\_mbrindp1.STATCODE := 'CA';  -- END IF;  -- ELSE  ----Condition TRANSFERFLG(ZPDATATXFLG) != 'Y'  -- obj\_mbrindp1.STATCODE := 'IF';  -- obj\_mbrindp1.dtetrm := C\_MAXDATE; -- MB15  -- END IF;  -- END IF;  -- END IF; -- MB15 -- IF\_ends  --END IF;  /\*dbms\_output.put\_line('TRIM(v\_refnump1)==>' || TRIM(v\_refnump1) ||  'obj\_mbrindp1.STATCODE=> ' ||  obj\_mbrindp1.STATCODE);\*/  --MB16 a\_checkpolst(TRIM(obj\_mbrindp1.refnum)) := obj\_mbrindp1.STATCODE;  --MB16 i\_statcode := a\_checkpolst(TRIM(obj\_mbrindp1.refnum));  --MB16 b\_checkpolst := a\_checkpolst;  --- MB16 END --  --- MB12 : MOD : Change for state code based on termination conditions : END  ---REFNUM NULL VALIDATION  -------------------------Flag based Changes :Start --------  IF ( getzigvalue.EXISTS(trim(v\_refnump1)) ) THEN  v\_zigvalue := getzigvalue(trim(v\_refnump1));  END IF;  -------------------------Flag based Changes :END --------  -- IF(i\_zprvaldYN != 'N') Then  IF TRIM(v\_refnump1) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z082;  t\_errorfieldp1(v\_errorcountp1) := 'REFNUM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z082);  t\_errorfieldvalp1(v\_errorcountp1) := trim(v\_refnump1);  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  GOTO insertzdoep1;  ELSE  ---Check for duplicate record IN ZDRPPF  /\* select count(RECIDXPOLICY)  into v\_isduplicatezdrppf  FROM VM1DTA.ZDRPPF  WHERE RTRIM(CHDRNUM) = TRIM(v\_refnump1);\*/  IF ( checkpoldup.EXISTS(trim(v\_refnump1)) ) THEN  -- IF v\_isduplicatezdrppf > 0 THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z001;  t\_errorfieldp1(v\_errorcountp1) := 'REFNUM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z001);  t\_errorfieldvalp1(v\_errorcountp1) := trim(v\_refnump1);  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  GOTO insertzdoep1;  END IF;  --Check for exist record IN ZDCLPF  /\* SELECT COUNT(\*) OVER(), r.zigvalue  into v\_isexistzdclpf, v\_zigvalue  FROM VM1DTA.ZDCLPF\_TEST r  WHERE TRIM(ZENTITY) = TRIM(v\_refnump1);\*/  /\* SELECT COUNT(\*) OVER(), r.zigvalue  into v\_isexistzdclpf, v\_zigvalue  FROM VM1DTA.ZDCLPF r  WHERE TRIM(ZENTITY) = TRIM(v\_refnump1);\*/  IF NOT ( getzigvalue.EXISTS(trim(v\_refnump1)) ) THEN  -- IF v\_isexistzdclpf < 1 THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z002;  t\_errorfieldp1(v\_errorcountp1) := 'REFNUM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z002);  t\_errorfieldvalp1(v\_errorcountp1) := trim(v\_refnump1);  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  ELSE  v\_zigvalue := getzigvalue(trim(v\_refnump1));  END IF;  ---Check for exist record IN TITDMGMBRINDP2  /\* SELECT COUNT(\*)  INTO v\_isexsittitdmgmbrindp2  FROM STAGEDBUSR.TITDMGMBRINDP2  WHERE TRIM(REFNUM) = TRIM(v\_refnump1);\*/  /\* IF NOT (checkrefnum.exists(TRIM(v\_refnumpseq))) THEN  -- IF v\_isexsittitdmgmbrindp2 < 1 THEN  v\_isAnyErrorp1 := 'Y';  v\_errorCountp1 := v\_errorCountp1 + 1;  t\_ercodep1(v\_errorCountp1) := C\_Z004;  t\_errorfieldp1(v\_errorCountp1) := 'REFNUM';  t\_errormsgp1(v\_errorCountp1) := o\_errortext(C\_Z004);  t\_errorfieldvalp1(v\_errorCountp1) := TRIM(v\_refnumpseq);  t\_errorprogramp1(v\_errorCountp1) := i\_scheduleName;  IF (v\_errorCountp1 = C\_ERRORCOUNTP1) THEN  GOTO insertzdoep1;  END IF;  -- END IF;  END IF;\*/  END IF;  --GPOLTYPE null validation  IF TRIM(obj\_mbrindp1.gpoltype) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z083;  t\_errorfieldp1(v\_errorcountp1) := 'GPOLTYPE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z083);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.gpoltype;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  --ZENDCDE null validation  IF TRIM(obj\_mbrindp1.zendcde) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z084;  t\_errorfieldp1(v\_errorcountp1) := 'ZENDCDE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z084);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zendcde;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  --ZCMPCODE null validation  IF TRIM(obj\_mbrindp1.zcmpcode) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z090;  t\_errorfieldp1(v\_errorcountp1) := 'ZCMPCODE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z090);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zcmpcode;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  --MPOLNUM null validation  IF obj\_mbrindp1.cnttypind = 'M' THEN  IF TRIM(obj\_mbrindp1.mpolnum) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z084;  t\_errorfieldp1(v\_errorcountp1) := 'MPOLNUM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z084);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.mpolnum;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  --MPOLNUM is not null validation  IF trim(obj\_mbrindp1.cnttypind) <> 'M' THEN  IF TRIM(obj\_mbrindp1.mpolnum) IS NOT NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z085;  t\_errorfieldp1(v\_errorcountp1) := 'MPOLNUM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z085);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.mpolnum;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---ZPOLPERD null validation  IF TRIM(obj\_mbrindp1.zpolperd) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z086;  t\_errorfieldp1(v\_errorcountp1) := 'ZPOLPERD';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z086);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zpolperd;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----ZMARGNFLG Value must be either ?Y? or ?N?  IF ( TRIM(obj\_mbrindp1.zmargnflg) IS NOT NULL AND ( ( trim(obj\_mbrindp1.zmargnflg) <> 'Y' ) AND ( trim(obj\_mbrindp1.  zmargnflg) <> 'N' ) ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_e315;  t\_errorfieldp1(v\_errorcountp1) := 'ZMARGNFLG';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_e315);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zmargnflg;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----ZDFCNCY Value must be either ?Y? or ?N?  IF ( TRIM(obj\_mbrindp1.zdfcncy) IS NOT NULL AND ( ( trim(obj\_mbrindp1.zdfcncy) <> 'Y' ) AND ( trim(obj\_mbrindp1.zdfcncy  ) <> 'N' ) ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_e315;  t\_errorfieldp1(v\_errorcountp1) := 'ZDFCNCY';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_e315);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zdfcncy;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----ZSOLCTFLG Value must be either ?Y? or ?N?  IF ( TRIM(obj\_mbrindp1.zsolctflg) IS NOT NULL AND ( ( trim(obj\_mbrindp1.zsolctflg) <> 'Y' ) AND ( trim(obj\_mbrindp1.  zsolctflg) <> 'N' ) ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_e315;  t\_errorfieldp1(v\_errorcountp1) := 'ZSOLCTFLG';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_e315);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zsolctflg;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ---CLTRELN null validation  IF TRIM(obj\_mbrindp1.cltreln) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z087;  t\_errorfieldp1(v\_errorcountp1) := 'CLTRELN';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z087);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.cltreln;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ---ZPLANCDE null validation  IF TRIM(obj\_mbrindp1.zplancde) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z088;  t\_errorfieldp1(v\_errorcountp1) := 'ZPLANCDE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z088);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zplancde;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----ZDCLITEM01 Value must be either ?Y? or ?N?  IF ( TRIM(obj\_mbrindp1.zdclitem01) IS NOT NULL AND ( ( trim(obj\_mbrindp1.zdclitem01) <> 'Y' ) AND ( trim(obj\_mbrindp1  .zdclitem01) <> 'N' ) ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_e315;  t\_errorfieldp1(v\_errorcountp1) := 'ZDCLITEM01';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_e315);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zdclitem01;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----ZDCLITEM02 Value must be either ?Y? or ?N?  IF ( TRIM(obj\_mbrindp1.zdclitem02) IS NOT NULL AND ( ( trim(obj\_mbrindp1.zdclitem02) <> 'Y' ) AND ( trim(obj\_mbrindp1  .zdclitem02) <> 'N' ) ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_e315;  t\_errorfieldp1(v\_errorcountp1) := 'ZDCLITEM02';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_e315);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zdclitem01;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----DTETRM Validation 1  v\_isdtetrmvalid := validate\_date(trim(obj\_mbrindp1.dtetrm));  IF ( ( trim(obj\_mbrindp1.statcode) = 'CA' ) AND ( ( TRIM(obj\_mbrindp1.dtetrm) IS NULL ) OR v\_isdtetrmvalid <> 'OK' ) )  THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_f623;  t\_errorfieldp1(v\_errorcountp1) := 'DTETRM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_f623);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.dtetrm;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----DTETRM Validation 1  IF ( ( trim(obj\_mbrindp1.statcode) <> 'CA' ) AND ( TRIM(obj\_mbrindp1.dtetrm) IS NOT NULL ) AND ( trim(obj\_mbrindp1.dtetrm  ) <> 0 ) AND ( trim(obj\_mbrindp1.dtetrm) <> c\_maxdate ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z089;  t\_errorfieldp1(v\_errorcountp1) := 'DTETRM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z089);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.dtetrm;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----STATCODE Validation  IF TRIM(obj\_mbrindp1.statcode) IS NULL THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_s008;  t\_errorfieldp1(v\_errorcountp1) := 'STATCODE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_s008);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.statcode;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  ELSE  IF ( trim(obj\_mbrindp1.statcode) <> 'PN' ) AND ( trim(obj\_mbrindp1.statcode) <> 'CA' ) AND ( trim(obj\_mbrindp1.statcode  ) <> 'XN' ) AND ( trim(obj\_mbrindp1.statcode) <> 'IF' ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_s008;  t\_errorfieldp1(v\_errorcountp1) := 'STATCODE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_s008);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.statcode;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ----------- New Validation For Newly Added Column ISSDATE -------------------------------------  IF trim(obj\_mbrindp1.statcode) = 'PN' THEN  n\_issdate := o\_defaultvalues('NOTSFROM');  ELSE  IF ( trim(obj\_mbrindp1.issdate) <> 99999999 AND TRIM(obj\_mbrindp1.issdate) IS NOT NULL ) THEN  n\_issdate := obj\_mbrindp1.issdate;  ELSE  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'ISSDATE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.issdate;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---------------Second part of validation -TITDMGMBRINDP1----------------------------------------  ---MPOLNUM validation  /\* IF TRIM(obj\_mbrindp1.mpolnum) IS NOT NULL THEN  select count(\*)  into v\_ismasterplmig  from GCHD  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.mpolnum);\*/  -- IF NOT (checkchdrnum.exists(TRIM(obj\_mbrindp1.mpolnum) || TRIM(i\_company))) THEN  IF TRIM(obj\_mbrindp1.mpolnum) IS NOT NULL THEN  IF NOT ( getmpol.EXISTS(trim(obj\_mbrindp1.mpolnum)) ) THEN  IF trim(v\_ismasterplmig) < 1 THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z003;  t\_errorfieldp1(v\_errorcountp1) := 'MPOLNUM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z003);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.mpolnum;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  END IF;  --ZSALECHNL validation  IF TRIM(obj\_mbrindp1.zsalechnl) IS NOT NULL THEN  IF NOT ( itemexist.EXISTS(trim('TQ9FW')  || trim(obj\_mbrindp1.zsalechnl)  || 1) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z005;  t\_errorfieldp1(v\_errorcountp1) := 'ZSALECHNL';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z005);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zsalechnl;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  --CLTRELN validation  IF TRIM(obj\_mbrindp1.cltreln) IS NOT NULL THEN  IF NOT ( itemexist.EXISTS(trim('T3584')  || trim(obj\_mbrindp1.cltreln)  || 9) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z006;  t\_errorfieldp1(v\_errorcountp1) := 'CLTRELN';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z006);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.cltreln;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  --ZTRXSTAT validation  IF TRIM(obj\_mbrindp1.ztrxstat) IS NOT NULL THEN  IF NOT ( itemexist.EXISTS(trim('TQ9FT')  || trim(obj\_mbrindp1.ztrxstat)  || 1) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z007;  t\_errorfieldp1(v\_errorcountp1) := 'ZTRXSTAT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z007);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.ztrxstat;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  --ZSTATRESN validation  IF TRIM(obj\_mbrindp1.zstatresn) IS NOT NULL THEN  IF NOT ( itemexist.EXISTS(trim('TQ9FU')  || trim(obj\_mbrindp1.zstatresn)  || 1) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z008;  t\_errorfieldp1(v\_errorcountp1) := 'ZSTATRESN';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z008);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zstatresn;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  --ZPLANCDE validation  IF TRIM(obj\_mbrindp1.zplancde) IS NOT NULL THEN  /\* select count(\*)  into v\_iszplancde  from ZSLPPF  where TRIM(ZSALPLAN) = obj\_mbrindp1.zplancde;\*/  IF NOT ( checksalplan.EXISTS(trim(obj\_mbrindp1.zplancde)) ) THEN  -- IF TRIM(v\_iszplancde) < 1 THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z009;  t\_errorfieldp1(v\_errorcountp1) := 'ZPLANCDE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z009);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zplancde;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  --ZENDCDE validation  /\* IF TRIM(obj\_mbrindp1.zendcde) IS NOT NULL THEN  select count(\*)  into v\_iszendcde  from ZENDRPF  where TRIM(ZENDCDE) = TRIM(obj\_mbrindp1.zendcde);\*/  IF NOT ( zendcde.EXISTS(trim(obj\_mbrindp1.zendcde)) ) THEN  IF trim(v\_iszendcde) < 1 THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z010;  t\_errorfieldp1(v\_errorcountp1) := 'ZENDCDE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z010);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zendcde;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  -- MB17 - MPS --  IF ( trim(obj\_mbrindp1.gpoltype) = 'FSH' ) THEN  IF ( ( TRIM(obj\_mbrindp1.crdtcard) IS NULL ) AND ( TRIM(obj\_mbrindp1.zenspcd01) IS NULL ) AND ( TRIM(obj\_mbrindp1  .zenspcd02) IS NULL ) AND ( TRIM(obj\_mbrindp1.zcifcode) IS NULL ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z011;  t\_errorfieldp1(v\_errorcountp1) := 'CRDTCARD';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z011);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.crdtcard;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  -- MB17 - MPS --  --At least one of the 3 fields (Credit Card No, Bank Account No and Endorser Specific Code) is mandatory. validation  IF ( trim(obj\_mbrindp1.gpoltype) <> 'FSH' ) THEN  IF ( ( TRIM(obj\_mbrindp1.crdtcard) IS NULL ) AND ( TRIM(obj\_mbrindp1.bnkacckey01) IS NULL ) AND ( TRIM(obj\_mbrindp1  .zenspcd01) IS NULL ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z011;  t\_errorfieldp1(v\_errorcountp1) := 'CRDTCARD';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z011);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.crdtcard;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  /\* --At least one of the 3 fields (Credit Card No, Bank Account No and Endorser Specific Code) is mandatory. validation  IF ((TRIM(obj\_mbrindp1.crdtcard) IS NULL) AND  (TRIM(obj\_mbrindp1.bnkacckey01) IS NULL) AND  (TRIM(obj\_mbrindp1.zenspcd01) IS NULL)) THEN  v\_isAnyErrorp1 := 'Y';  v\_errorCountp1 := INSERT\_ZDOE(v\_prefix,  i\_scheduleNumber,  TRIM(v\_refKeyp1),  C\_Z011,  'BNKACCKEY01',  obj\_mbrindp1.bnkacckey01,  NULL,  'E');  IF (v\_errorCountp1 = C\_ERRORCOUNTP1) THEN  GOTO insertzdoep1;  END IF;  END IF;  --At least one of the 3 fields (Credit Card No, Bank Account No and Endorser Specific Code) is mandatory. validation  IF ((TRIM(obj\_mbrindp1.crdtcard) IS NULL) AND  (TRIM(obj\_mbrindp1.bnkacckey01) IS NULL) AND  (TRIM(obj\_mbrindp1.zenspcd01) IS NULL)) THEN  v\_isAnyErrorp1 := 'Y';  v\_errorCountp1 := INSERT\_ZDOE(v\_prefix,  i\_scheduleNumber,  TRIM(v\_refKeyp1),  C\_Z011,  'ZENSPCD01',  obj\_mbrindp1.zenspcd01,  NULL,  'E');  IF (v\_errorCountp1 = C\_ERRORCOUNTP1) THEN  GOTO insertzdoep1;  END IF;  END IF;\*/  --ZCMPCODE validation  /\*IF TRIM(obj\_mbrindp1.zcmpcode) IS NOT NULL THEN  select count(\*)  into v\_iszcmpcode  from ZCPNPF  where TRIM(ZCMPCODE) = TRIM(obj\_mbrindp1.zcmpcode);\*/  IF NOT ( campcode.EXISTS(trim(obj\_mbrindp1.zcmpcode)) ) THEN  IF trim(v\_iszcmpcode) < 1 THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z021;  t\_errorfieldp1(v\_errorcountp1) := 'ZCMPCODE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z021);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zcmpcode;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ----CNTTYPIND Value must be either ?M? or ?I?  IF ( TRIM(obj\_mbrindp1.cnttypind) IS NOT NULL AND ( ( trim(obj\_mbrindp1.cnttypind) <> 'M' ) AND ( trim(obj\_mbrindp1.  cnttypind) <> 'I' ) ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z012;  t\_errorfieldp1(v\_errorcountp1) := 'CNTTYPIND';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z012);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.cnttypind;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----EFFDATE Validation  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.effdate));  IF ( ( TRIM(obj\_mbrindp1.effdate) IS NULL ) OR ( v\_iseffdate <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'EFFDATE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.effdate;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----DOCRCVDT Validation  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.docrcvdt));  IF ( ( TRIM(obj\_mbrindp1.docrcvdt) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'DOCRCVDT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.docrcvdt;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----HPROPDTE Validation  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.hpropdte));  IF ( ( TRIM(obj\_mbrindp1.hpropdte) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'HPROPDTE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.hpropdte;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ----ZANNCLDT Validation  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.zanncldt));  IF ( ( TRIM(obj\_mbrindp1.zanncldt) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'ZANNCLDT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zanncldt;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ---PTDATE validation  IF ( trim(obj\_mbrindp1.statcode) <> 'PN' ) THEN  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.ptdate));  IF ( ( TRIM(obj\_mbrindp1.ptdate) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'PTDATE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.ptdate;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---BTDATE validation  IF ( trim(obj\_mbrindp1.statcode) <> 'PN' ) AND ( trim(obj\_mbrindp1.ztrxstat) <> 'RJ' ) THEN  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.btdate));  IF ( ( TRIM(obj\_mbrindp1.btdate) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'BTDATE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.btdate;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---ITR-3 ZWAITPEDT validation  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.zwaitpedt));  IF ( ( TRIM(obj\_mbrindp1.zwaitpedt) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'ZWAITPEDT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zwaitpedt;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ---ITR-3 ZCONVINDPOL must be blank  IF ( TRIM(obj\_mbrindp1.zconvindpol) IS NOT NULL AND ( trim(obj\_mbrindp1.cnttypind) <> 'M' ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z093;  t\_errorfieldp1(v\_errorcountp1) := 'ZCONVINDPO';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z093);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zconvindpol;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  /\*---ITR-3 ZCONVINDPOL is invalid ---As removed in TSD  IF (TRIM(obj\_mbrindp1.ZCONVINDPOL) IS NOT NULL AND  (TRIM(obj\_mbrindp1.cnttypind) = 'M')) THEN  select count(\*)  into v\_ismasterplmig  from GCHD  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.ZCONVINDPOL);  IF TRIM(v\_ismasterplmig) < 1 THEN  v\_isAnyErrorp1 := 'Y';  v\_errorCountp1 := v\_errorCountp1 + 1;  t\_ercodep1(v\_errorCountp1) := C\_Z120;  t\_errorfieldp1(v\_errorCountp1) := 'ZCONVINDPOL';  t\_errormsgp1(v\_errorCountp1) := o\_errortext(C\_Z120);  t\_errorfieldvalp1(v\_errorCountp1) := obj\_mbrindp1.ZCONVINDPOL;  t\_errorprogramp1(v\_errorCountp1) := i\_scheduleName;  IF (v\_errorCountp1 = C\_ERRORCOUNTP1) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;\*/  --ITR-3 OLDPOLNUM Old Pol No must be blank-  IF ( TRIM(obj\_mbrindp1.oldpolnum) IS NOT NULL AND ( trim(obj\_mbrindp1.cnttypind) <> 'I' ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z095;  t\_errorfieldp1(v\_errorcountp1) := 'OLDPOLNUM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z095);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.oldpolnum;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  /\* --ITR-3 OLDPOLNUM Invalid Old Policy No. ---As removed in TSD  IF (TRIM(obj\_mbrindp1.OLDPOLNUM) IS NOT NULL AND  (TRIM(obj\_mbrindp1.cnttypind) = 'I')) THEN  select count(\*)  into v\_ismasterplmig  from GCHD  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.OLDPOLNUM);  IF TRIM(v\_ismasterplmig) < 1 THEN  v\_isAnyErrorp1 := 'Y';  v\_errorCountp1 := v\_errorCountp1 + 1;  t\_ercodep1(v\_errorCountp1) := C\_Z094;  t\_errorfieldp1(v\_errorCountp1) := 'OLDPOLNUM';  t\_errormsgp1(v\_errorCountp1) := o\_errortext(C\_Z094);  t\_errorfieldvalp1(v\_errorCountp1) := obj\_mbrindp1.OLDPOLNUM;  t\_errorprogramp1(v\_errorCountp1) := i\_scheduleName;  IF (v\_errorCountp1 = C\_ERRORCOUNTP1) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;\*/  --ITR-3 TREFNUM TREFNUM is invalid  ---SIT bug changes  /\* IF (TRIM(obj\_mbrindp1.TREFNUM) IS NOT NULL) THEN  select count(\*)  into v\_ismasterplmig  from GCHD  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.TREFNUM);  IF TRIM(v\_ismasterplmig) < 1 THEN  v\_isAnyErrorp1 := 'Y';  v\_errorCountp1 := v\_errorCountp1 + 1;  t\_ercodep1(v\_errorCountp1) := C\_Z121;  t\_errorfieldp1(v\_errorCountp1) := 'TREFNUM';  t\_errormsgp1(v\_errorCountp1) := o\_errortext(C\_Z121);  t\_errorfieldvalp1(v\_errorCountp1) := obj\_mbrindp1.TREFNUM;  t\_errorprogramp1(v\_errorCountp1) := i\_scheduleName;  IF (v\_errorCountp1 = C\_ERRORCOUNTP1) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;\*/  --ITR-3 TREFNUM ZCONVINDPOL is invalid  -- MB9 [START]  -- IF (TRIM(obj\_mbrindp1.TREFNUM) IS NOT NULL AND  -- (TRIM(obj\_mbrindp1.ZCONVINDPOL) IS NOT NULL)) THEN  -- v\_isAnyErrorp1 := 'Y';  -- v\_errorCountp1 := v\_errorCountp1 + 1;  -- t\_ercodep1(v\_errorCountp1) := C\_Z120;  -- t\_errorfieldp1(v\_errorCountp1) := 'TREFNUM';  -- t\_errormsgp1(v\_errorCountp1) := o\_errortext(C\_Z120);  -- t\_errorfieldvalp1(v\_errorCountp1) := obj\_mbrindp1.TREFNUM;  -- t\_errorprogramp1(v\_errorCountp1) := i\_scheduleName;  -- IF (v\_errorCountp1 = C\_ERRORCOUNTP1) THEN  -- GOTO insertzdoep1;  -- END IF;  -- END IF;  -- MB9 [END]  --ITR-3 TREFNUM Old Pol No must be blank  IF ( TRIM(obj\_mbrindp1.trefnum) IS NOT NULL AND ( TRIM(obj\_mbrindp1.oldpolnum) IS NOT NULL ) ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z095;  t\_errorfieldp1(v\_errorcountp1) := 'TREFNUM';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z095);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.trefnum;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ---ITR-3 ZPOLTDATE validation Invalid Date  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.zpoltdate));  IF ( ( TRIM(obj\_mbrindp1.zpoltdate) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'ZPOLTDATE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zpoltdate;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ---ITR-3 ZPOLTDATE validation Outside policy period  IF ( ( TRIM(obj\_mbrindp1.zpoltdate) IS NOT NULL ) AND ( trim(obj\_mbrindp1.zpoltdate) <> 0 ) AND ( trim(obj\_mbrindp1.  zpoltdate) <> c\_maxdate ) ) THEN  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  IF ( ( trim(obj\_mbrindp1.zpoltdate) < obj\_mbrindp1.effdate ) OR ( trim(obj\_mbrindp1.zpoltdate) > v\_temp\_crdate ) )  THEN  /\*\*\*\* MB4 : MOD : condition change due to new requirement : END \*\*\*\*/  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z096;  t\_errorfieldp1(v\_errorcountp1) := 'ZPOLTDATE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z096);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zpoltdate;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---ITR-3 ZPGPFRDT validation Invalid Date  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.zpgpfrdt));  IF ( ( TRIM(obj\_mbrindp1.zpgpfrdt) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'ZPGPFRDT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zpgpfrdt;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ---ITR-3 ZPGPFRDT validation Outside policy period  IF ( ( TRIM(obj\_mbrindp1.zpgpfrdt) IS NOT NULL ) AND ( trim(obj\_mbrindp1.zpgpfrdt) <> 0 ) AND ( trim(obj\_mbrindp1.zpgpfrdt  ) <> c\_maxdate ) ) THEN  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  IF ( ( trim(obj\_mbrindp1.zpgpfrdt) < obj\_mbrindp1.effdate ) OR ( trim(obj\_mbrindp1.zpgpfrdt) > v\_temp\_crdate ) ) THEN  /\*\*\*\* MB4 : MOD : condition change due to new requirement : END \*\*\*\*/  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z096;  t\_errorfieldp1(v\_errorcountp1) := 'ZPGPFRDT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z096);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zpgpfrdt;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---ITR-3 ZPGPTODT validation Invalid Date  v\_iseffdate := validate\_date(trim(obj\_mbrindp1.zpgptodt));  IF ( ( TRIM(obj\_mbrindp1.zpgptodt) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'ZPGPTODT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zpgptodt;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  ---ITR-3 ZPGPTODT validation From date > To date  IF ( TRIM(obj\_mbrindp1.zpgpfrdt) IS NOT NULL ) THEN  IF ( ( TRIM(obj\_mbrindp1.zpgptodt) IS NULL ) OR ( trim(obj\_mbrindp1.zpgptodt) < obj\_mbrindp1.zpgpfrdt ) ) THEN  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z097;  t\_errorfieldp1(v\_errorcountp1) := 'ZPGPTODT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z097);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zpgptodt;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---ITR-3 ZPGPTODT validation Outside policy period  IF ( ( TRIM(obj\_mbrindp1.zpgptodt) IS NOT NULL ) AND ( trim(obj\_mbrindp1.zpgptodt) <> 0 ) AND ( trim(obj\_mbrindp1.zpgptodt  ) <> c\_maxdate ) ) THEN  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  IF ( ( trim(obj\_mbrindp1.zpgptodt) < obj\_mbrindp1.effdate ) OR ( trim(obj\_mbrindp1.zpgptodt) > v\_temp\_crdate ) ) THEN  /\*\*\*\* MB4 : MOD : condition change due to new requirement : END \*\*\*\*/  v\_isanyerrorp1 := 'Y'; --MB12  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z096;  t\_errorfieldp1(v\_errorcountp1) := 'ZPGPTODT';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z096);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.zpgptodt;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---MB12 : MOD :BTDATE allow As MAX DATE only for CA state : START  ---BTDATE validation  IF ( trim(obj\_mbrindp1.statcode) <> 'CA' ) THEN  IF ( trim(obj\_mbrindp1.btdate) = c\_maxdate ) THEN  v\_isanyerrorp1 := 'Y';  v\_errorcountp1 := v\_errorcountp1 + 1;  t\_ercodep1(v\_errorcountp1) := c\_z013;  t\_errorfieldp1(v\_errorcountp1) := 'BTDATE';  t\_errormsgp1(v\_errorcountp1) := o\_errortext(c\_z013);  t\_errorfieldvalp1(v\_errorcountp1) := obj\_mbrindp1.btdate;  t\_errorprogramp1(v\_errorcountp1) := i\_schedulename;  IF ( v\_errorcountp1 = c\_errorcountp1 ) THEN  GOTO insertzdoep1;  END IF;  END IF;  END IF;  ---MB12 : MOD :BTDATE allow As MAX DATE only for CA state : END  << insertzdoep1 >> i\_zdoe\_infop1.i\_tablename := i\_zdoe\_infop1.i\_tablename;  IF ( v\_isanyerrorp1 = 'Y' ) THEN  IF TRIM(t\_ercodep1(1)) IS NOT NULL THEN  i\_zdoe\_infop1.i\_indic := c\_error;  i\_zdoe\_infop1.i\_error01 := t\_ercodep1(1);  i\_zdoe\_infop1.i\_errormsg01 := t\_errormsgp1(1);  i\_zdoe\_infop1.i\_errorfield01 := t\_errorfieldp1(1);  i\_zdoe\_infop1.i\_fieldvalue01 := t\_errorfieldvalp1(1);  i\_zdoe\_infop1.i\_errorprogram01 := t\_errorprogramp1(1);  END IF;  IF TRIM(t\_ercodep1(2)) IS NOT NULL THEN  i\_zdoe\_infop1.i\_indic := c\_error;  i\_zdoe\_infop1.i\_error02 := t\_ercodep1(2);  i\_zdoe\_infop1.i\_errormsg02 := t\_errormsgp1(2);  i\_zdoe\_infop1.i\_errorfield02 := t\_errorfieldp1(2);  i\_zdoe\_infop1.i\_fieldvalue02 := t\_errorfieldvalp1(2);  i\_zdoe\_infop1.i\_errorprogram02 := t\_errorprogramp1(2);  END IF;  IF TRIM(t\_ercodep1(3)) IS NOT NULL THEN  i\_zdoe\_infop1.i\_indic := c\_error;  i\_zdoe\_infop1.i\_error03 := t\_ercodep1(3);  i\_zdoe\_infop1.i\_errormsg03 := t\_errormsgp1(3);  i\_zdoe\_infop1.i\_errorfield03 := t\_errorfieldp1(3);  i\_zdoe\_infop1.i\_fieldvalue03 := t\_errorfieldvalp1(3);  i\_zdoe\_infop1.i\_errorprogram03 := t\_errorprogramp1(3);  END IF;  IF TRIM(t\_ercodep1(4)) IS NOT NULL THEN  i\_zdoe\_infop1.i\_indic := c\_error;  i\_zdoe\_infop1.i\_error04 := t\_ercodep1(4);  i\_zdoe\_infop1.i\_errormsg04 := t\_errormsgp1(4);  i\_zdoe\_infop1.i\_errorfield04 := t\_errorfieldp1(4);  i\_zdoe\_infop1.i\_fieldvalue04 := t\_errorfieldvalp1(4);  i\_zdoe\_infop1.i\_errorprogram04 := t\_errorprogramp1(4);  END IF;  IF TRIM(t\_ercodep1(5)) IS NOT NULL THEN  i\_zdoe\_infop1.i\_indic := c\_error;  i\_zdoe\_infop1.i\_error05 := t\_ercodep1(5);  i\_zdoe\_infop1.i\_errormsg05 := t\_errormsgp1(5);  i\_zdoe\_infop1.i\_errorfield05 := t\_errorfieldp1(5);  i\_zdoe\_infop1.i\_fieldvalue05 := t\_errorfieldvalp1(5);  i\_zdoe\_infop1.i\_errorprogram05 := t\_errorprogramp1(5);  END IF;  pkg\_dm\_common\_operations.insertintozdoe(i\_zdoe\_info => i\_zdoe\_infop1);  CONTINUE skiprecord;  END IF;  IF ( v\_isanyerrorp1 = 'N' ) THEN  i\_zdoe\_infop1.i\_indic := 'S';  pkg\_dm\_common\_operations.insertintozdoe(i\_zdoe\_info => i\_zdoe\_infop1);  END IF;  -- END IF;  IF ( ( trim(i\_zprvaldyn) = 'N' ) AND ( trim(v\_isanyerrorp1) = 'N' ) ) THEN  IF ( itemexist.EXISTS(trim('TQ9FK')  || trim(obj\_mbrindp1.gpoltype)  || 1) ) THEN  -- v\_template := itemexist(obj\_mbrindp1.gpoltype).template;  v\_template := itemexist(trim('TQ9FK')  || trim(obj\_mbrindp1.gpoltype)  || trim('1')).template;  /\* select \*  into obj\_dfpopf  from DFPOPF  where TRIM(template) = TRIM(v\_template); \*/  IF ( getdfpo.EXISTS(trim(v\_template)) ) THEN  obj\_dfpopf := getdfpo(trim(v\_template));  END IF;  END IF;  SELECT  seq\_chdrpf.NEXTVAL  INTO v\_seq\_chdrpf  FROM  dual;  obj\_gchd.unique\_number := v\_seq\_chdrpf;  obj\_gchd.chdrcoy := i\_company;  obj\_gchd.chdrnum := substr(trim(obj\_mbrindp1.refnum), 1, 8);  obj\_gchd.chdrpfx := o\_defaultvalues('CHDRPFX');  obj\_gchd.cnttype := obj\_mbrindp1.gpoltype;  obj\_gchd.cownpfx := o\_defaultvalues('COWNPFX');  obj\_gchd.cowncoy := i\_fsucocompany;  /\* IF v\_isexistzdclpf > 0 THEN  SELECT zigvalue  INTO v\_zigvalue1  FROM VM1DTA.ZDCLPF  WHERE TRIM(ZENTITY) = TRIM(v\_refnump1);  END IF; \*/  obj\_gchd.cownnum := v\_zigvalue;  obj\_gchd.statcode := obj\_mbrindp1.statcode;  IF ( trim(obj\_mbrindp1.statcode) = 'PN' ) THEN  obj\_gchd.pndate := obj\_mbrindp1.effdate;  ELSE  obj\_gchd.pndate := c\_maxdate;  END IF;  -- obj\_gchd.SUBSFLG := obj\_dfpopf.Subsflg;  obj\_gchd.subsflg := c\_space; --MB7  --- 17/02/2018 After Pre-SIT execution  /\* IF ((TRIM(obj\_mbrindp1.zanncldt) IS NULL) OR  ((TRIM(obj\_mbrindp1.zanncldt) = C\_MAXDATE)) OR  ((TRIM(obj\_mbrindp1.zanncldt) = C\_ZERO))) THEN  obj\_gchd.OCCDATE := obj\_mbrindp1.effdate;  ELSE  obj\_gchd.OCCDATE := TRIM(v\_zcovcmdt);  END IF;\*/  obj\_gchd.occdate := obj\_mbrindp1.effdate;  obj\_gchd.hrskind := obj\_dfpopf.hrskind;  /\* SELECT TRIM(SUBSTR(utl\_raw.cast\_to\_varchar2(GENAREA), 0, 3))  into v\_currency  FROM ITEMPF  WHERE RTRIM(ITEMTABL) = 'T9775'  and TRIM(ITEMITEM) = TRIM(i\_branch)  AND RTRIM(ITEMCOY) = TRIM(i\_company); \*/  IF ( itemexist.EXISTS(trim('T9775')  || trim(i\_branch)  || 1) ) THEN  -- v\_currency := itemexist(i\_branch).currency;  v\_currency := itemexist(trim('T9775')  || trim(i\_branch)  || trim('1')).currency;  END IF;  obj\_gchd.cntcurr := trim(v\_currency);  obj\_gchd.billcurr := trim(v\_currency);  -- obj\_gchd.TAKOVRFLG := obj\_dfpopf.takovrflg;  obj\_gchd.takovrflg := c\_space; --MB7  obj\_gchd.gprnltyp := obj\_dfpopf.gprnltyp;  obj\_gchd.gprmnths := obj\_dfpopf.gprmnths;  obj\_gchd.rnlnotto := obj\_dfpopf.rnlnotto;  obj\_gchd.srcebus := o\_defaultvalues('SRCEBUS');  obj\_gchd.coysrvac := obj\_dfpopf.coysrvac;  obj\_gchd.mrksrvac := obj\_dfpopf.mrksrvac;  obj\_gchd.desppfx := c\_space;  obj\_gchd.despcoy := c\_space;  obj\_gchd.despnum := c\_space;  obj\_gchd.polschpflg := o\_defaultvalues('POLSCHPFLG');  obj\_gchd.btdate := obj\_mbrindp1.btdate;  obj\_gchd.adjdate := c\_maxdate;  obj\_gchd.ptdate := obj\_mbrindp1.ptdate;  obj\_gchd.ptdateab := c\_maxdate;  obj\_gchd.tranlused := o\_defaultvalues('TRANLUSED');  obj\_gchd.lmbrno := o\_defaultvalues('LMBRNO');  obj\_gchd.lheadno := o\_defaultvalues('LHEADNO');  IF ( trim(obj\_mbrindp1.statcode) = 'CA' ) THEN  obj\_gchd.effdcldt := obj\_mbrindp1.dtetrm;  ELSE  obj\_gchd.effdcldt := c\_maxdate;  END IF;  obj\_gchd.servunit := o\_defaultvalues('SERVUNIT');  IF ( trim(obj\_mbrindp1.statcode) = 'PN' ) THEN  obj\_gchd.pntrcde := o\_defaultvalues('PNTRCDE');  ELSE  obj\_gchd.pntrcde := c\_space;  END IF;  obj\_gchd.procid := c\_space;  obj\_gchd.tranid := concat('QPAD', TO\_CHAR(SYSDATE, 'YYMMDDHHMM'));  --IF (TRIM(obj\_mbrindp1.statcode) = 'CA') THEN  -- obj\_gchd.TRANNO := v\_seqnoincr;  --ELSE  -- obj\_gchd.TRANNO := v\_seqno;  --END IF;  obj\_gchd.tranno := v\_seqnoincr;  obj\_gchd.validflag := o\_defaultvalues('VALIDFLAG');  obj\_gchd.specind := c\_space;  obj\_gchd.taxflag := obj\_dfpopf.taxflag;  obj\_gchd.agedef := obj\_dfpopf.agedef;  obj\_gchd.termage := obj\_dfpopf.termage;  obj\_gchd.personcov := obj\_dfpopf.personcov;  obj\_gchd.enrolltyp := obj\_dfpopf.enrolltyp;  obj\_gchd.splitsubs := obj\_dfpopf.splitsubs;  obj\_gchd.avlisu := c\_space;  -- obj\_gchd.MPLPFX := o\_defaultvalues('CHDRPFX');  obj\_gchd.mplpfx := c\_space; --MB7  obj\_gchd.mplcoy := i\_company;  IF ( trim(obj\_mbrindp1.cnttypind) = 'I' ) THEN  obj\_gchd.mplnum := c\_space;  ELSE  obj\_gchd.mplnum := obj\_mbrindp1.mpolnum;  END IF;  obj\_gchd.usrprf := i\_usrprf;  obj\_gchd.jobnm := i\_schedulename;  obj\_gchd.datime := CAST ( SYSDATE AS TIMESTAMP );  obj\_gchd.igrasp := c\_space;  obj\_gchd.iexplain := c\_space;  obj\_gchd.idate := c\_zero;  obj\_gchd.midjoin := c\_space;  obj\_gchd.cvisaind := c\_space;  obj\_gchd.covernt := c\_space;  obj\_gchd.cntiss := c\_zero;  -- obj\_gchd.REPNUM := C\_ZERO;  obj\_gchd.repnum := c\_space; --MB7  obj\_gchd.reptype := c\_space;  obj\_gchd.payrcoy := c\_space;  obj\_gchd.payrnum := c\_space;  obj\_gchd.payrpfx := c\_space;  IF ( TRIM(obj\_mbrindp1.oldpolnum) IS NOT NULL ) THEN  obj\_gchd.zprvchdr := obj\_mbrindp1.oldpolnum;  ELSE  obj\_gchd.zprvchdr := obj\_mbrindp1.trefnum;  END IF;  --- 17/02/2018 After Pre-SIT execution  obj\_gchd.currfrom := c\_zero;  obj\_gchd.currto := c\_zero;  -- obj\_gchd.TRANLUSED := v\_seqnoincr;  obj\_gchd.tranlused := obj\_gchd.tranno; --MB12 : TRANNO == TRANLUSED  obj\_gchd.zrwnlage := o\_defaultvalues('ZRWNLAGE'); --MB7 Need to change TQ9Q9  obj\_gchd.quoteno := c\_zero; --MB7 new added  obj\_gchd.schmno := c\_space; --MB7 new added  obj\_gchd.rtgrante := c\_space; --MB7 new added  obj\_gchd.rtgrantedate := c\_zero; --MB7 new added  obj\_gchd.cpiincrind := c\_space; --MB7 new added  obj\_gchd.superflag := c\_space; --MB7 new added  IF obj\_gchd.statcode = 'XN' THEN  obj\_gchd.ptdate := c\_maxdate;  END IF;  --INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO GCHD VALUES obj\_gchd;  ---MB18  gchdindex := gchdindex + 1;  gchd\_list.extend;  gchd\_list(gchdindex) := obj\_gchd;  ---MB18  ------Insert Into IG table "GCHD" END -----  ---Insert into IG table "GCHPPF" BEGIN (pq9ho.updateGchppf())---  SELECT  zagptid,  zpolcls  INTO  v\_zagptid,  v\_zpolcls  FROM  zcpnpf  WHERE  TRIM(zcmpcode) = TRIM(obj\_mbrindp1.zcmpcode);  SELECT  seq\_gchppf.NEXTVAL  INTO v\_seq\_gchppf  FROM  dual;  obj\_gchppf.unique\_number := v\_seq\_gchppf;  obj\_gchppf.chdrcoy := i\_company;  obj\_gchppf.chdrnum := substr(trim(obj\_mbrindp1.refnum), 1, 8);  obj\_gchppf.exbrknm := obj\_dfpopf.exbrknm;  obj\_gchppf.exundnm := obj\_dfpopf.exundnm;  obj\_gchppf.brksrvac := obj\_dfpopf.brksrvac;  obj\_gchppf.refno := obj\_dfpopf.refno;  obj\_gchppf.mbrdata := obj\_dfpopf.mbrdata;  obj\_gchppf.admnrule := obj\_dfpopf.admnrule;  obj\_gchppf.defplandi := obj\_dfpopf.defplandi;  obj\_gchppf.defclmpye := obj\_dfpopf.defclmpye;  obj\_gchppf.empgrp := obj\_dfpopf.empgrp;  obj\_gchppf.inwinctyp := obj\_dfpopf.inwinctyp;  obj\_gchppf.areacod := obj\_dfpopf.areacod;  obj\_gchppf.industry := obj\_dfpopf.industry;  obj\_gchppf.majormet := obj\_dfpopf.majormet;  obj\_gchppf.bulkind := obj\_dfpopf.bulkind;  obj\_gchppf.ffeewhom := c\_space;  obj\_gchppf.prodmix := obj\_dfpopf.prodmix;  obj\_gchppf.feelvl := obj\_dfpopf.feelvl;  obj\_gchppf.ctbeffdt := obj\_dfpopf.ctbeffdt;  obj\_gchppf.exbfml := obj\_dfpopf.exbfml;  obj\_gchppf.exbldays := obj\_dfpopf.exbldays;  obj\_gchppf.ctbfml := obj\_dfpopf.ctbfml;  obj\_gchppf.ctbndays := obj\_dfpopf.ctbndays;  obj\_gchppf.efais := obj\_dfpopf.efais;  obj\_gchppf.efadp := obj\_dfpopf.efadp;  obj\_gchppf.norem := obj\_dfpopf.norem;  obj\_gchppf.fstrmfml := obj\_dfpopf.fstrmfml;  obj\_gchppf.fstrmday := obj\_dfpopf.fstrmday;  obj\_gchppf.sndrmfml := obj\_dfpopf.sndrmfml;  obj\_gchppf.sndrmday := obj\_dfpopf.sndrmday;  obj\_gchppf.trdrmfml := obj\_dfpopf.trdrmfml;  obj\_gchppf.trdrmday := obj\_dfpopf.trdrmday;  obj\_gchppf.mbridfld := obj\_dfpopf.mbridfld;  obj\_gchppf.exbduedt := c\_maxdate;  obj\_gchppf.ctbduedt := c\_maxdate;  obj\_gchppf.lstexbfr := c\_maxdate;  obj\_gchppf.lstexbto := c\_maxdate;  obj\_gchppf.lstexbto := c\_maxdate;  obj\_gchppf.lstctbto := c\_maxdate;  obj\_gchppf.lstebpdt := c\_maxdate;  obj\_gchppf.fstrmpdt := c\_maxdate;  obj\_gchppf.sndrmpdt := c\_maxdate;  obj\_gchppf.trdrmpdt := c\_maxdate;  IF ( TRIM(obj\_mbrindp1.mpolnum) IS NOT NULL ) THEN  /\* select POLANV  into v\_polanv  from VM1DTA.GCHPPF  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.mpolnum)  and TRIM(CHDRCOY) = TRIM(i\_company);\*/  obj\_getmpol := getmpol(trim(obj\_mbrindp1.mpolnum));  v\_polanv := obj\_getmpol.polanv;  /\* IF (getpolanv.exists(TRIM(obj\_mbrindp1.mpolnum) || TRIM(i\_company ))) THEN  v\_polanv := getpolanv(TRIM(obj\_mbrindp1.mpolnum) || TRIM(i\_company)) ;  END IF;\*/  obj\_gchppf.polanv := v\_polanv;  END IF;  obj\_gchppf.ctbrule := obj\_dfpopf.ctbrule;  obj\_gchppf.acblrule := obj\_dfpopf.acblrule;  obj\_gchppf.fmcrule := obj\_dfpopf.fmcrule;  obj\_gchppf.swtranno := c\_zero;  obj\_gchppf.feewho := obj\_dfpopf.feewho;  obj\_gchppf.zsrcebus := c\_space;  obj\_gchppf.calcmthd := obj\_dfpopf.calcmthd;  obj\_gchppf.agebasis := obj\_dfpopf.agebasis;  obj\_gchppf.fcllvl := obj\_dfpopf.fcllvl;  obj\_gchppf.prmpyopt := obj\_dfpopf.prmpyopt;  obj\_gchppf.prmbrlvl := obj\_dfpopf.prmbrlvl;  obj\_gchppf.tolrule := obj\_dfpopf.tolrule;  obj\_gchppf.certinfm := obj\_dfpopf.certinfm;  obj\_gchppf.fmc2rule := obj\_dfpopf.fmc2rule;  obj\_gchppf.lmbrpfx := c\_space;  obj\_gchppf.loybnflg := obj\_dfpopf.loybnflg;  obj\_gchppf.swcflg := obj\_dfpopf.swcflg;  obj\_gchppf.autornw := obj\_dfpopf.autornw;  obj\_gchppf.gaplpfx := obj\_dfpopf.gaplpfx;  obj\_gchppf.nmlvar := obj\_dfpopf.nmlvar;  obj\_gchppf.extfmly := obj\_dfpopf.extfmly;  obj\_gchppf.pinfdte := c\_maxdate;  obj\_gchppf.cashless := o\_defaultvalues('CASHLESS');  obj\_gchppf.location := c\_space;  obj\_gchppf.sublocn := c\_space;  obj\_gchppf.ttdate := c\_maxdate;  obj\_gchppf.jobnm := i\_schedulename;  obj\_gchppf.usrprf := i\_usrprf;  obj\_gchppf.datime := CAST ( SYSDATE AS TIMESTAMP );  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  obj\_gchppf.zpenddt := v\_temp\_crdate;  /\*\*\*\* MB4 : MOD : condition change due to new requirement : END \*\*\*\*/  /\* IF (TRIM(obj\_mbrindp1.mpolnum) IS NOT NULL) THEN  select ZPLANCLS, Zcolmcls  into v\_zplancls, v\_zcolmcls  from VM1DTA.GCHPPF  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.mpolnum)  and CHDRCOY = TRIM(i\_company);  obj\_gchppf.ZPLANCLS := v\_zplancls;  ELSE  obj\_gchppf.ZPLANCLS := 'PP';  END IF; \*/  -- obj\_gchppf.PLANCLASSIFICATION := 'a';  -------- MB2: get value of ZCOLMCLS and ZPLANCLS --------  IF ( TRIM(obj\_mbrindp1.mpolnum) IS NOT NULL ) THEN  obj\_getmpol := getmpol(trim(obj\_mbrindp1.mpolnum));  v\_zplancls := obj\_getmpol.zplancls;  v\_zcolmcls := obj\_getmpol.zcolmcls;  v\_zplancls\_new := v\_zplancls;  ELSE  v\_zplancls\_new := 'PP';  /\* Use ZENDCDE and read ZENDRPF, and get Fact-House (zendrpf.getZfacthus())  if Fact-house not blank, use Fact-house and read t-table T3684 and get bnkacctyp.  set ZCOLMCLS = t3684rec.bnkacctyp.toString() \*/  v\_zfacthus := facthouse(trim(obj\_mbrindp1.zendcde));  IF v\_zfacthus IS NOT NULL THEN  IF itemexist.EXISTS(trim('T3684')  || trim(v\_zfacthus)  || 9) THEN  v\_zcolmcls := itemexist(trim('T3684')  || trim(v\_zfacthus)  || 9).bnkacctyp;  END IF;  END IF;  END IF;  -------- MB2: get value of ZCOLMCLS and ZPLANCLS --------  obj\_gchppf.zplancls := v\_zplancls\_new;  --obj\_gchppf.ZAPLFOD := C\_MAXDATE; -- obj\_gchppf.APPLICATIONFORMOUTPUT := 'a'; -- MB8  obj\_gchppf.zaplfod := 0; -- obj\_gchppf.APPLICATIONFORMOUTPUT := 'a'; -- MB8  obj\_gchppf.zgporipcls := v\_zpolcls; -- obj\_gchppf.GROUPCLASSIFICATION := 'a';  obj\_gchppf.zendcde := obj\_mbrindp1.zendcde; -- obj\_gchppf.ENDORSERCODE := 'a';  ---03/05/18 SIT bug fix  -- obj\_gchppf.Zccode := obj\_mbrindp1.zcmpcode; -- obj\_gchppf.CAMPAIGNCODE := 'a';  obj\_gchppf.petname := c\_space;  obj\_gchppf.lstctbfr := c\_maxdate;  obj\_gchppf.optautornw := c\_space;  obj\_gchppf.ocallvsa := c\_space;  obj\_gchppf.zpolperd := obj\_mbrindp1.zpolperd;  obj\_gchppf.zcolmcls := v\_zcolmcls;  IF ( trim(obj\_mbrindp1.cnttypind) = 'M' ) THEN  obj\_gchppf.zconvindpol := obj\_mbrindp1.zconvindpol;  ELSE  obj\_gchppf.zconvindpol := c\_space;  END IF;  IF ( TRIM(obj\_mbrindp1.zpoltdate) IS NOT NULL ) THEN  obj\_gchppf.zpoltdate := obj\_mbrindp1.zpoltdate;  ELSE  obj\_gchppf.zpoltdate := c\_maxdate;  END IF;  IF ( TRIM(obj\_mbrindp1.zpgpfrdt) IS NOT NULL ) THEN  obj\_gchppf.zpgpfrdt := obj\_mbrindp1.zpgpfrdt;  ELSE  obj\_gchppf.zpgpfrdt := c\_maxdate;  END IF;  IF ( TRIM(obj\_mbrindp1.zpgptodt) IS NOT NULL ) THEN  obj\_gchppf.zpgptodt := obj\_mbrindp1.zpgptodt;  ELSE  obj\_gchppf.zpgptodt := c\_maxdate;  END IF;  obj\_gchppf.sinstno := obj\_mbrindp1.sinstno;  --SIT BUG FIx  obj\_gchppf.znbmnage := o\_defaultvalues('ZNBMNAGE');  obj\_gchppf.matage := c\_zero; --MB7 new addedd  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO GCHPPF VALUES obj\_gchppf;  ---MB18  gchppfindex := gchppfindex + 1;  gchppf\_list.extend;  gchppf\_list(gchppfindex) := obj\_gchppf;  ---MB18  ---Insert into IG table "GCHPPF" END ---  ---Insert into IG table "GCHIPF" BEGIN (pq9ho.updateGchipf())--  SELECT  seq\_gchipf.NEXTVAL  INTO v\_seq\_gchipf  FROM  dual;  obj\_gchipf.unique\_number := v\_seq\_gchipf;  obj\_gchipf.chdrcoy := i\_company;  obj\_gchipf.chdrnum := substr(trim(obj\_mbrindp1.refnum), 1, 8);  obj\_gchipf.effdate := obj\_mbrindp1.effdate;  obj\_gchipf.ccdate := obj\_mbrindp1.effdate;  obj\_gchipf.zpstddt := obj\_mbrindp1.effdate;  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  obj\_gchipf.crdate := v\_temp\_crdate;  /\*\*\*\* MB4 : MOD : condition change due to new requirement : END \*\*\*\*/  -- obj\_gchipf.CRDATE := obj\_mbrindp1.crdate;  -- obj\_gchipf.PRVBILFLG := obj\_dfpopf.prvbilflg;  obj\_gchipf.prvbilflg := c\_space; --MB7  IF ( trim(v\_zplancls) = 'FP' ) THEN  obj\_gchipf.billfreq := '00'; --MB14  ELSE  obj\_gchipf.billfreq := obj\_dfpopf.billfreq;  END IF; ---MB7  IF ( trim(v\_zplancls) = 'FP' ) THEN  obj\_gchipf.gadjfreq := '00'; --MB14  ELSE  obj\_gchipf.gadjfreq := obj\_dfpopf.gadjfreq;  END IF; --MB7  obj\_gchipf.payrpfx := c\_space;  obj\_gchipf.payrcoy := c\_space;  obj\_gchipf.payrnum := c\_space;  obj\_gchipf.agntpfx := o\_defaultvalues('AGNTPFX');  obj\_gchipf.agntcoy := i\_fsucocompany;  IF ( trim(obj\_mbrindp1.cnttypind) <> 'I' ) THEN  /\* select zagptid, zpolcls  into v\_zagptid, v\_zpolcls  from Zcpnpf  where TRIM(ZCMPCODE) = TRIM(obj\_mbrindp1.zcmpcode);\*/  /\* select ZAGPTNUM  into v\_zagptnum  from GCHIPF  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.mpolnum)  AND EFFDATE <= TRIM(obj\_mbrindp1.effdate)  and ROWNUM = 1  order by tranno DESC;\*/  obj\_getmpol := getmpol(trim(obj\_mbrindp1.mpolnum));  v\_zagptnum := obj\_getmpol.zagptnum;  SELECT  admnoper01,  gagntsel01,  admnoper02,  gagntsel02,  admnoper03,  gagntsel03,  admnoper04,  gagntsel04,  admnoper05,  gagntsel05  INTO  v\_admnoper01,  v\_gagntsel01,  v\_admnoper02,  v\_gagntsel02,  v\_admnoper03,  v\_gagntsel03,  v\_admnoper04,  v\_gagntsel04,  v\_admnoper05,  v\_gagntsel05  FROM  zagppf  WHERE  TRIM(zagptnum) = v\_zagptnum  AND TRIM(zagptpfx) = 'AP'  AND TRIM(zagptcoy) = i\_company;  IF trim(v\_admnoper01) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel01;  END IF;  IF trim(v\_admnoper02) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel02;  END IF;  IF trim(v\_admnoper03) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel03;  END IF;  IF trim(v\_admnoper04) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel04;  END IF;  IF trim(v\_admnoper05) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel05;  END IF;  ELSE  /\* select zagptid, zpolcls  into v\_zagptid, v\_zpolcls  from Zcpnpf  where TRIM(ZCMPCODE) = TRIM(obj\_mbrindp1.zcmpcode);\*/  SELECT  admnoper01,  gagntsel01,  admnoper02,  gagntsel02,  admnoper03,  gagntsel03,  admnoper04,  gagntsel04,  admnoper05,  gagntsel05  INTO  v\_admnoper01,  v\_gagntsel01,  v\_admnoper02,  v\_gagntsel02,  v\_admnoper03,  v\_gagntsel03,  v\_admnoper04,  v\_gagntsel04,  v\_admnoper05,  v\_gagntsel05  FROM  zagppf  WHERE  TRIM(zagptnum) = v\_zagptid  AND TRIM(zagptpfx) = 'AP'  AND TRIM(zagptcoy) = i\_company;  IF trim(v\_admnoper01) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel01;  END IF;  IF trim(v\_admnoper02) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel02;  END IF;  IF trim(v\_admnoper03) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel03;  END IF;  IF trim(v\_admnoper04) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel04;  END IF;  IF trim(v\_admnoper05) = 'Y' THEN  obj\_gchipf.agntnum := v\_gagntsel05;  END IF;  END IF;  obj\_gchipf.zagptnum := v\_zagptid;  obj\_gchipf.cntbranch := i\_branch;  -- obj\_gchipf.STCA := C\_SPACE;  obj\_gchipf.stca := 'LF'; --MB7  obj\_gchipf.stcb := c\_space;  obj\_gchipf.stcc := c\_space;  obj\_gchipf.stcd := c\_space;  obj\_gchipf.stce := c\_space;  IF ( trim(obj\_mbrindp1.cnttypind) = 'M' ) THEN  /\* select ZPLANCLS  into v\_zplancls  from VM1DTA.GCHPPF  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.mpolnum)  and CHDRCOY = TRIM(i\_company);\*/  obj\_getmpol := getmpol(trim(obj\_mbrindp1.mpolnum));  v\_zplancls := obj\_getmpol.zplancls;  IF ( trim(v\_zplancls) = 'FP' ) THEN  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  --obj\_gchipf.BTDATENR := v\_temp\_crdate;  obj\_gchipf.btdatenr := to\_number(TO\_CHAR(TO\_DATE(v\_temp\_crdate, 'YYYYMMDD') + 1, 'yyyymmdd'));  /\*\*\*\* MB4 : MOD : condition change due to new requirement : END \*\*\*\*/  ELSE  v\_btdatenr := datconoperation('MONTH', obj\_gchipf.ccdate);  -- IF (TRIM(obj\_mbrindp1.btdate) <> C\_MAXDATE) THEN  -- --v\_btdatenr := DATCONOPERATION('DAY', obj\_mbrindp1.btdate);  -- v\_btdatenr := DATCONOPERATION('MONTH', obj\_gchipf.CCDATE); -- MB3: GCHIPF.CCDATE + 1 month  -- -- dbms\_output.put\_line('obj\_gchipf.BTDATENR : v\_btdatenr =>' || v\_btdatenr);  -- obj\_gchipf.BTDATENR := v\_btdatenr;  -- ELSE  -- obj\_gchipf.BTDATENR := obj\_mbrindp1.btdate;  -- END IF;  END IF;  END IF;  obj\_gchipf.nrisdate := c\_maxdate;  obj\_gchipf.crate := obj\_dfpopf.crate;  obj\_gchipf.ternmprm := c\_zero;  obj\_gchipf.surgschmv := obj\_dfpopf.surgschmv;  obj\_gchipf.areacdemv := obj\_dfpopf.areacdemv;  obj\_gchipf.medprvdr := obj\_dfpopf.medprvdr;  obj\_gchipf.spsmbr := obj\_dfpopf.spsmbr;  obj\_gchipf.childmbr := obj\_dfpopf.childmbr;  obj\_gchipf.spsmed := obj\_dfpopf.spsmed;  obj\_gchipf.childmed := obj\_dfpopf.childmed;  obj\_gchipf.termid := i\_vrcmtermid;  --obj\_gchipf.USER\_T := i\_vrcmuser; -- obj\_gchipf.USER := 'a';  obj\_gchipf.user\_t := 36; --MB7  obj\_gchipf.trdt := i\_trdt;  obj\_gchipf.trtm := i\_vrcmtime;  obj\_gchipf.tranno := 1;  --IF (TRIM(obj\_mbrindp1.statcode) = 'CA') THEN  -- obj\_gchipf.TRANNO := v\_seqnoincr;  --ELSE  -- obj\_gchipf.TRANNO := v\_seqno;  --END IF;  obj\_gchipf.bankcode := obj\_dfpopf.bankcode;  obj\_gchipf.billchnl := obj\_dfpopf.billchnl;  obj\_gchipf.mandref := c\_space;  obj\_gchipf.rimthvcd := obj\_dfpopf.rimthvcd;  -- obj\_gchipf.PRMRVWDT := C\_MAXDATE;  obj\_gchipf.prmrvwdt := c\_zero; --MB7  obj\_gchipf.appltyp := obj\_dfpopf.appltyp;  obj\_gchipf.riind := obj\_dfpopf.riind;  -- obj\_gchipf.POLBREAK := o\_defaultvalues('POLBREAK');  obj\_gchipf.polbreak := c\_space; --MB7  obj\_gchipf.cftype := c\_space;  obj\_gchipf.lmtdrl := c\_space;  obj\_gchipf.cflimit := c\_zero;  obj\_gchipf.nofclaim := c\_zero;  -- obj\_gchipf.TPA := C\_SPACE;--MB7 comment out as setting null  obj\_gchipf.wkladrt := c\_zero;  obj\_gchipf.wklcmrt := c\_zero;  obj\_gchipf.nofmbr := c\_zero;  -- obj\_gchipf.ZCOLMCLS := 'a'; --still in discussion  obj\_gchipf.usrprf := i\_usrprf;  obj\_gchipf.jobnm := i\_schedulename;  obj\_gchipf.datime := CAST ( SYSDATE AS TIMESTAMP );  obj\_gchipf.ecnv := o\_defaultvalues('ECNV');  obj\_gchipf.cvntype := o\_defaultvalues('CVNTYPE');  obj\_gchipf.covernt := c\_space;  /\* SELECT TRIM(SUBSTR(utl\_raw.cast\_to\_varchar2(GENAREA), 0, 5)),  TRIM(SUBSTR(utl\_raw.cast\_to\_varchar2(GENAREA), 6, 5))  into v\_timech01, v\_timech02  FROM ITEMPF  WHERE RTRIM(ITEMTABL) = 'TQ9GX'  and TRIM(ITEMITEM) = 'SHIT902'  AND RTRIM(ITEMCOY) = TRIM(i\_company); \*/  IF ( itemexist.EXISTS(trim('TQ9GX')  || trim('SHIT902')  || 1) ) THEN  -- v\_timech01 := itemexist('SHIT902').timech01;  -- v\_timech02 := itemexist('SHIT902').timech02;  v\_timech01 := itemexist(trim('TQ9GX')  || trim('SHIT902')  || trim('1')).timech01;  v\_timech02 := itemexist(trim('TQ9GX')  || trim('SHIT902')  || trim('1')).timech02;  END IF;  obj\_gchipf.timech01 := v\_timech01;  obj\_gchipf.timech02 := v\_timech02;  obj\_gchipf.tpaflg := c\_space;  obj\_gchipf.docrcdte := obj\_mbrindp1.docrcvdt;  obj\_gchipf.insstdte := obj\_mbrindp1.effdate;  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  /\*\*\*\* MB12 : MOD : INSENDTE recalculation CRDATE+1 : START \*\*\*\*/  SELECT  to\_number(TO\_CHAR(TO\_DATE(obj\_gchipf.crdate, 'yyyymmdd') + 1, 'yyyymmdd'))  INTO v\_insendate  FROM  dual;  obj\_gchipf.insendte := v\_insendate;  /\*\*\*\* MB12 : MOD : INSENDTE recalculation CRDATE+1 : END \*\*\*\*/  /\*\*\*\* MB4 : MOD : condition change due to new requirement : START \*\*\*\*/  obj\_gchipf.zsolctflg := obj\_mbrindp1.zsolctflg;  --- 17/02/2018 After Pre-SIT execution  obj\_gchipf.cownnum := v\_zigvalue;  ---03/05/18 SIT bug fix  obj\_gchipf.zcmpcode := obj\_mbrindp1.zcmpcode;  obj\_gchipf.zpenddt := v\_temp\_crdate;  --INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO GCHIPF VALUES obj\_gchipf;  ---MB18  gchipfindex := gchipfindex + 1;  gchipf\_list.extend;  gchipf\_list(gchipfindex) := obj\_gchipf;  ---MB18  ---Insert into IG table "GCHIPF" END ---  ---Insert into IG table "GMHDPF" BEGIN (pq9ho.updateGmhdpf()) ---  SELECT  seq\_gmhdpf.NEXTVAL  INTO v\_seq\_gmhdpf  FROM  dual;  obj\_gmhdpf.unique\_number := v\_seq\_gmhdpf;  obj\_gmhdpf.chdrcoy := i\_company;  obj\_gmhdpf.chdrnum := substr(trim(obj\_mbrindp1.refnum), 1, 8);  obj\_gmhdpf.mbrno := o\_defaultvalues('MBRNO');  obj\_gmhdpf.dpntno := o\_defaultvalues('DPNTNO');  --- 17/02/2018 After Pre-SIT execution  IF ( trim(obj\_mbrindp1.statcode) = 'CA' ) THEN  obj\_gmhdpf.dtetrm := obj\_mbrindp1.dtetrm;  ELSE  obj\_gmhdpf.dtetrm := c\_maxdate;  END IF;  --obj\_gmhdpf.DTETRM := C\_MAXDATE;  obj\_gmhdpf.reasontrm := o\_defaultvalues('REASONTRM');  -- obj\_gmhdpf.PNDATE := obj\_mbrindp1.docrcvdt;  obj\_gmhdpf.pndate := c\_maxdate;  obj\_gmhdpf.clntpfx := o\_defaultvalues('CLNTPFX');  obj\_gmhdpf.fsuco := i\_fsucocompany;  obj\_gmhdpf.clntnum := v\_zigvalue;  obj\_gmhdpf.headcnt := o\_defaultvalues('HEADCNT');  obj\_gmhdpf.dteatt := obj\_mbrindp1.effdate;  -- obj\_gmhdpf.MEDEVD := C\_SPACE;--MB7 comment out as setting null  obj\_gmhdpf.reln := o\_defaultvalues('RELN');  obj\_gmhdpf.fauwdt := c\_maxdate;  obj\_gmhdpf.ldpntno := o\_defaultvalues('LDPNTNO');  obj\_gmhdpf.termid := i\_vrcmtermid;  obj\_gmhdpf.trdt := i\_trdt;  obj\_gmhdpf.trtm := i\_vrcmtime;  IF ( trim(obj\_mbrindp1.statcode) = 'CA' ) THEN  obj\_gmhdpf.tranno := v\_seqnoincr;  ELSE  obj\_gmhdpf.tranno := v\_seqno;  END IF;  obj\_gmhdpf.client := v\_zigvalue;  -- obj\_gmhdpf.TERMRSNCD := o\_defaultvalues('TERMRSNCD');--MB7 comment out as setting null  -- obj\_gmhdpf.REFIND := o\_defaultvalues('REFIND');--MB7 comment out as setting null  -- obj\_gmhdpf.EMPNO := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.SMOKEIND := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.OCCPCLAS := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.ETHORG := C\_SPACE;--MB7 comment out as setting null  obj\_gmhdpf.gheight := c\_zero;  obj\_gmhdpf.gweight := c\_zero;  -- obj\_gmhdpf.PAYMMETH := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.DEFCLMPYE := C\_SPACE;--MB7 comment out as setting null  obj\_gmhdpf.prvpoldt := c\_maxdate;  -- obj\_gmhdpf.DEPT := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.NEWOLDCL := C\_SPACE;--MB7 comment out as setting null  -- dbms\_output.put\_line('v\_zigvalue =>' || v\_zigvalue);  /\* select CLTDOB  into v\_cltdob  from CLNTPF  where TRIM(CLNTNUM) = TRIM(v\_zigvalue);\*/  /\* obj\_getclnt := getclntinfo(TRIM(v\_zigvalue));  v\_cltdob := obj\_getclnt.cltdob;\*/  --for pga v\_cltdob := clntdob(TRIM(v\_zigvalue));  ---MB18:  v\_cltdob := obj\_mbrindp1.cltdob;  ---MB18:  --select Cltdob into v\_cltdob from vm1dta.CLNTPF where clntnum = v\_zigvalue;  v\_datconage := datediff('YEAR', v\_cltdob, obj\_mbrindp1.effdate);  -- dbms\_output.put\_line('v\_datconage =>' || v\_datconage);  obj\_gmhdpf.age := v\_datconage;  obj\_gmhdpf.ordob := c\_zero;  -- obj\_gmhdpf.STATCODE := obj\_mbrindp1.statcode;  -- obj\_gmhdpf.STATCODE := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.BANKACCKEY := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.APPLICNO := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.CERTNO := C\_SPACE;--MB7 comment out as setting null  obj\_gmhdpf.medcmpdt := c\_maxdate;  /\* IF (TRIM(obj\_mbrindp1.statcode) = 'IF') THEN  obj\_gmhdpf.INFORCE := o\_defaultvalues('INFORCE');  ELSE  obj\_gmhdpf.INFORCE := C\_SPACE;  END IF; \*/ --MB7 comment out as setting null  -- obj\_gmhdpf.WEIGHTUNIT := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.HEIGHTUNIT := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhdpf.MBRTYPC := C\_SPACE;--MB7 comment out as setting null  obj\_gmhdpf.sifact := c\_bigdecimal\_default1;  -- obj\_gmhdpf.RDYPROC := C\_SPACE; --MB7 comment out as setting null  /\* IF (TRIM(obj\_mbrindp1.statcode) = 'IF') THEN  obj\_gmhdpf.INSUFFMN := o\_defaultvalues('INSUFFMN');  ELSE  obj\_gmhdpf.INSUFFMN := C\_SPACE;  END IF;\*/  -- obj\_gmhdpf.INSUFFMN := C\_SPACE;  obj\_gmhdpf.insuffmn := 'Y'; --MB7  -- obj\_gmhdpf.DPNTTYPE := C\_SPACE; MB7 comment out as setting null  obj\_gmhdpf.usrprf := i\_usrprf;  obj\_gmhdpf.jobnm := i\_schedulename;  obj\_gmhdpf.datime := CAST ( SYSDATE AS TIMESTAMP );  obj\_gmhdpf.user\_t := i\_vrcmuser;  obj\_gmhdpf.zanncldt := obj\_mbrindp1.zanncldt;  obj\_gmhdpf.zcpnscde := obj\_mbrindp1.zcmpcode || obj\_mbrindp1.zcpnscde02;  -- obj\_gmhdpf.ZCPNSCDE := C\_SPACE;  obj\_gmhdpf.zsalechnl := obj\_mbrindp1.zsalechnl;  obj\_gmhdpf.cltreln := o\_defaultvalues('RELN');  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO GMHDPF VALUES obj\_gmhdpf;  ---MB18  gmhdpfindex := gmhdpfindex + 1;  gmhdpf\_list.extend;  gmhdpf\_list(gmhdpfindex) := obj\_gmhdpf;  ---MB18  --------insert into IG table "GMOVPF" BEGIN(pq9ho.updateGmhdpf())----  -- IF (TRIM(obj\_mbrindp1.statcode) <> 'PN') THEN Previous code  IF ( trim(obj\_mbrindp1.statcode) = 'XN' ) THEN  v\_olddta1 := obj\_gmhdpf.chdrcoy  || obj\_gmhdpf.chdrnum  || obj\_gmhdpf.mbrno  || obj\_gmhdpf.dpntno  || c\_maxdate  || obj\_gmhdpf.reasontrm  || obj\_gmhdpf.clntpfx  || obj\_gmhdpf.fsuco  || obj\_gmhdpf.clntnum  || obj\_gmhdpf.headcnt  || obj\_gmhdpf.dteatt  || obj\_gmhdpf.medevd  || obj\_gmhdpf.reln  || obj\_gmhdpf.fauwdt  || obj\_gmhdpf.ldpntno  || obj\_gmhdpf.termid  ||  -- obj\_gmhdpf.UserT ||  obj\_gmhdpf.trdt  || obj\_gmhdpf.trtm  || obj\_gmhdpf.tranno  || obj\_gmhdpf.pndate  || obj\_gmhdpf.client  || obj\_gmhdpf.termrsncd  || obj\_gmhdpf.refind  || obj\_gmhdpf.empno  || obj\_gmhdpf.smokeind  || obj\_gmhdpf.occpclas  || obj\_gmhdpf.ethorg  || obj\_gmhdpf.gheight  || obj\_gmhdpf.gweight  || obj\_gmhdpf.paymmeth  || obj\_gmhdpf.defclmpye  || obj\_gmhdpf.prvpoldt  || obj\_gmhdpf.dept  || obj\_gmhdpf.newoldcl  || obj\_gmhdpf.age  || obj\_gmhdpf.ordob  || 'IF'  || obj\_gmhdpf.bankacckey  || obj\_gmhdpf.applicno  || obj\_gmhdpf.certno  || obj\_gmhdpf.medcmpdt  || obj\_gmhdpf.inforce  || obj\_gmhdpf.weightunit  || obj\_gmhdpf.heightunit  || obj\_gmhdpf.mbrtypc  || obj\_gmhdpf.sifact  || obj\_gmhdpf.rdyproc  || obj\_gmhdpf.insuffmn  || obj\_gmhdpf.dpnttype  || obj\_gmhdpf.usrprf  || obj\_gmhdpf.jobnm  || obj\_gmhdpf.datime  || obj\_gmhdpf.zcpnscde  || obj\_gmhdpf.zsalechnl;  --obj\_gmhdpf.Zanncldt();  v\_refkey := obj\_gmhdpf.chdrcoy  || obj\_gmhdpf.chdrnum  || obj\_gmhdpf.mbrno  || obj\_gmhdpf.dpntno;  obj\_gmovpf1.chdrcoy := obj\_gmhdpf.chdrcoy;  obj\_gmovpf1.chdrnum := obj\_gmhdpf.chdrnum;  obj\_gmovpf1.effdate := obj\_mbrindp1.effdate;  obj\_gmovpf1.tranno := 1;  obj\_gmovpf1.user\_t := obj\_gmhdpf.user\_t;  obj\_gmovpf1.trdt := obj\_gmhdpf.trdt;  obj\_gmovpf1.trtm := obj\_gmhdpf.trtm;  obj\_gmovpf1.termid := obj\_gmhdpf.termid;  obj\_gmovpf1.batccoy := i\_company;  obj\_gmovpf1.batcbrn := i\_branch;  obj\_gmovpf1.batcactyr := i\_acctyear;  obj\_gmovpf1.batcactmn := i\_acctmonth;  IF ( trim(obj\_mbrindp1.statcode) = 'CA' ) THEN  obj\_gmovpf1.batctrcde := 'T913';  ELSE  obj\_gmovpf1.batctrcde := 'T903';  END IF;  obj\_gmovpf1.batcbatch := c\_space;  obj\_gmovpf1.olddta := v\_olddta1;  obj\_gmovpf1.newdta := '';  obj\_gmovpf1.funccode := c\_funcdeaddmbr;  obj\_gmovpf1.refkey := v\_refkey;  obj\_gmovpf1.rfmt := 'GMHDREC';  obj\_gmovpf1.chgtype := '';  obj\_gmovpf1.riprior := ' ';  obj\_gmovpf1.onpreflg := '';  obj\_gmovpf1.fachold := '';  obj\_gmovpf1.planvflg := '';  obj\_gmovpf1.usrprf := i\_usrprf;  obj\_gmovpf1.jobnm := i\_schedulename;  obj\_gmovpf1.datime := CAST ( SYSDATE AS TIMESTAMP );  SELECT  seq\_gmovpf.NEXTVAL  INTO v\_seq\_gmovpf  FROM  dual;  obj\_gmovpf1.unique\_number := v\_seq\_gmovpf;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO GMOVPF VALUES obj\_gmovpf1;  ---MB18  gmovpf1index := gmovpf1index + 1;  gmovpf1\_list.extend;  gmovpf1\_list(gmovpf1index) := obj\_gmovpf1;  ---MB18  END IF;  /\* IF (TRIM(obj\_mbrindp1.statcode) = 'CA') THEN  v\_olddta1 := obj\_gmhdpf.Chdrcoy || obj\_gmhdpf.Chdrnum ||  obj\_gmhdpf.Mbrno || obj\_gmhdpf.Dpntno || C\_MAXDATE ||  obj\_gmhdpf.Reasontrm || obj\_gmhdpf.Clntpfx ||  obj\_gmhdpf.Fsuco || obj\_gmhdpf.Clntnum ||  obj\_gmhdpf.Headcnt || obj\_gmhdpf.Dteatt ||  obj\_gmhdpf.Medevd || obj\_gmhdpf.Reln ||  obj\_gmhdpf.Fauwdt || obj\_gmhdpf.Ldpntno ||  obj\_gmhdpf.Termid ||  -- obj\_gmhdpf.UserT ||  obj\_gmhdpf.Trdt || obj\_gmhdpf.Trtm ||  obj\_gmhdpf.Tranno || obj\_gmhdpf.Pndate ||  obj\_gmhdpf.Client || obj\_gmhdpf.Termrsncd ||  obj\_gmhdpf.Refind || obj\_gmhdpf.Empno ||  obj\_gmhdpf.Smokeind || obj\_gmhdpf.Occpclas ||  obj\_gmhdpf.Ethorg || obj\_gmhdpf.Gheight ||  obj\_gmhdpf.Gweight || obj\_gmhdpf.Paymmeth ||  obj\_gmhdpf.Defclmpye || obj\_gmhdpf.Prvpoldt ||  obj\_gmhdpf.Dept || obj\_gmhdpf.Newoldcl ||  obj\_gmhdpf.Age || obj\_gmhdpf.Ordob || 'IF' ||  obj\_gmhdpf.Bankacckey || obj\_gmhdpf.Applicno ||  obj\_gmhdpf.Certno || obj\_gmhdpf.Medcmpdt ||  obj\_gmhdpf.Inforce || obj\_gmhdpf.Weightunit ||  obj\_gmhdpf.Heightunit || obj\_gmhdpf.Mbrtypc ||  obj\_gmhdpf.Sifact || obj\_gmhdpf.Rdyproc ||  obj\_gmhdpf.Insuffmn || obj\_gmhdpf.Dpnttype ||  obj\_gmhdpf.Usrprf || obj\_gmhdpf.Jobnm ||  obj\_gmhdpf.Datime || obj\_gmhdpf.Zcpnscde ||  obj\_gmhdpf.Zsalechnl;  --obj\_gmhdpf.Zanncldt();  v\_newdta1 := obj\_gmhdpf.Chdrcoy || obj\_gmhdpf.Chdrnum ||  obj\_gmhdpf.Mbrno || obj\_gmhdpf.Dpntno ||  obj\_gmhdpf.Dtetrm || obj\_gmhdpf.Reasontrm ||  obj\_gmhdpf.Clntpfx || obj\_gmhdpf.Fsuco ||  obj\_gmhdpf.Clntnum || obj\_gmhdpf.Headcnt ||  obj\_gmhdpf.Dteatt || obj\_gmhdpf.Medevd ||  obj\_gmhdpf.Reln || obj\_gmhdpf.Fauwdt ||  obj\_gmhdpf.Ldpntno || obj\_gmhdpf.Termid ||  obj\_gmhdpf.user\_t || obj\_gmhdpf.Trdt ||  obj\_gmhdpf.Trtm || obj\_gmhdpf.Tranno ||  obj\_gmhdpf.Pndate || obj\_gmhdpf.Client ||  obj\_gmhdpf.Termrsncd || obj\_gmhdpf.Refind ||  obj\_gmhdpf.Empno || obj\_gmhdpf.Smokeind ||  obj\_gmhdpf.Occpclas || obj\_gmhdpf.Ethorg ||  obj\_gmhdpf.Gheight || obj\_gmhdpf.Gweight ||  obj\_gmhdpf.Paymmeth || obj\_gmhdpf.Defclmpye ||  obj\_gmhdpf.Prvpoldt || obj\_gmhdpf.Dept ||  obj\_gmhdpf.Newoldcl || obj\_gmhdpf.Age ||  obj\_gmhdpf.Ordob || obj\_gmhdpf.Statcode ||  obj\_gmhdpf.Bankacckey || obj\_gmhdpf.Applicno ||  obj\_gmhdpf.Certno || obj\_gmhdpf.Medcmpdt ||  obj\_gmhdpf.Inforce || obj\_gmhdpf.Weightunit ||  obj\_gmhdpf.Heightunit || obj\_gmhdpf.Mbrtypc ||  obj\_gmhdpf.Sifact || obj\_gmhdpf.Rdyproc ||  obj\_gmhdpf.Insuffmn || obj\_gmhdpf.Dpnttype ||  obj\_gmhdpf.Usrprf || obj\_gmhdpf.Jobnm ||  obj\_gmhdpf.Datime || obj\_gmhdpf.Zcpnscde ||  obj\_gmhdpf.Zsalechnl || obj\_gmhdpf.Zanncldt;  v\_refkey := obj\_gmhdpf.Chdrcoy + obj\_gmhdpf.Chdrnum +  obj\_gmhdpf.Mbrno + obj\_gmhdpf.Dpntno;  obj\_gmovpf1.Chdrcoy := obj\_gmhdpf.Chdrcoy;  obj\_gmovpf1.Chdrnum := obj\_gmhdpf.Chdrnum;  obj\_gmovpf1.Effdate := obj\_mbrindp1.effdate;  obj\_gmovpf1.Tranno := 2;  obj\_gmovpf1.User\_t := obj\_gmhdpf.User\_t;  obj\_gmovpf1.Trdt := obj\_gmhdpf.Trdt;  obj\_gmovpf1.Trtm := obj\_gmhdpf.Trtm;  obj\_gmovpf1.Termid := obj\_gmhdpf.Termid;  obj\_gmovpf1.Batccoy := i\_company;  obj\_gmovpf1.Batcbrn := i\_branch;  obj\_gmovpf1.Batcactyr := i\_acctYear;  obj\_gmovpf1.Batcactmn := i\_acctMonth;  IF (TRIM(obj\_mbrindp1.statcode) = 'CA') THEN  obj\_gmovpf1.Batctrcde := 'T913';  ELSE  obj\_gmovpf1.Batctrcde := 'T913';  END IF;  obj\_gmovpf1.Batcbatch := C\_SPACE;  obj\_gmovpf1.Olddta := v\_olddta1;  obj\_gmovpf1.Newdta := v\_newdta1;  obj\_gmovpf1.Funccode := C\_FUNCDETRMBR;  obj\_gmovpf1.Refkey := v\_refkey;  obj\_gmovpf1.Rfmt := 'GMHDREC';  obj\_gmovpf1.Chgtype := '';  obj\_gmovpf1.Riprior := ' ';  obj\_gmovpf1.Onpreflg := '';  obj\_gmovpf1.Fachold := '';  obj\_gmovpf1.Planvflg := '';  obj\_gmovpf1.Usrprf := i\_usrprf;  obj\_gmovpf1.Jobnm := i\_scheduleName;  obj\_gmovpf1.Datime := CAST(sysdate AS TIMESTAMP);  Insert into GMOVPF values obj\_gmovpf1;  END IF;\*/  --------insert into IG table "GMOVPF" END (pq9ho.updateGmhdpf())----  ---Insert into IG table "GMHDPF" END ---  ---Insert into IG table "GMHIPF" BEGIN (pq9ho.updateGmhipf()) ---  SELECT  seq\_gmhipf.NEXTVAL  INTO v\_seq\_gmhipf  FROM  dual;  obj\_gmhipf.unique\_number := v\_seq\_gmhipf;  obj\_gmhipf.chdrcoy := i\_company;  obj\_gmhipf.chdrnum := substr(trim(obj\_mbrindp1.refnum), 1, 8);  obj\_gmhipf.mbrno := o\_defaultvalues('MBRNO');  obj\_gmhipf.dpntno := o\_defaultvalues('DPNTNO');  obj\_gmhipf.effdate := obj\_mbrindp1.effdate;  IF ( trim(obj\_mbrindp1.statcode) = 'CA' ) THEN  obj\_gmhipf.dtetrm := obj\_mbrindp1.dtetrm;  ELSE  obj\_gmhipf.dtetrm := c\_maxdate;  END IF;  -- obj\_gmhipf.DTETRM := C\_MAXDATE;  -- obj\_gmhipf.SUBSCOY := C\_SPACE; --MB7 comment out as setting null  -- obj\_gmhipf.SUBSNUM := C\_SPACE; --MB7 comment out as setting null  -- obj\_gmhipf.OCCPCODE := C\_SPACE; --MB7 comment out as setting null  obj\_gmhipf.salary := c\_bigdecimal\_default;  obj\_gmhipf.dteapp := c\_maxdate;  obj\_gmhipf.sbstdl := c\_space;  obj\_gmhipf.termid := i\_vrcmtermid;  obj\_gmhipf.trdt := i\_trdt;  obj\_gmhipf.trtm := i\_vrcmtime;  IF ( trim(obj\_mbrindp1.statcode) = 'CA' ) THEN  obj\_gmhipf.tranno := v\_seqnoincr;  ELSE  obj\_gmhipf.tranno := v\_seqno;  END IF;  -- obj\_gmhipf.FUPFLG := C\_SPACE; --MB7 comment out as setting null  -- obj\_gmhipf.CLIENT := v\_zigvalue; --MB7 comment out as setting null  obj\_gmhipf.personcov := o\_defaultvalues('PERSONCOV');  obj\_gmhipf.mlvlplan := o\_defaultvalues('MLVLPLAN');  -- obj\_gmhipf.CLNTCOY := i\_fsucocompany;  obj\_gmhipf.clntcoy := c\_space; --MB7  -- obj\_gmhipf.PCCCLNT := C\_SPACE;--MB7 comment out as setting null  -- obj\_gmhipf.APCCCLNT := C\_SPACE;--MB7 comment out as setting null  obj\_gmhipf.earning := c\_bigdecimal\_default;  obj\_gmhipf.ctbprcnt := c\_bigdecimal\_default;  obj\_gmhipf.ctbamt := c\_bigdecimal\_default;  obj\_gmhipf.usrprf := i\_usrprf;  obj\_gmhipf.jobnm := i\_schedulename;  obj\_gmhipf.datime := CAST ( SYSDATE AS TIMESTAMP );  obj\_gmhipf.ztrxstat := obj\_mbrindp1.ztrxstat;  obj\_gmhipf.quoteno := c\_space;  obj\_gmhipf.zcpnscde := obj\_mbrindp1.zcpnscde02; -- obj\_gmhipf.ZCPNSCDE := obj\_mbrindp1.zcmpcode;  obj\_gmhipf.zdclitem01 := obj\_mbrindp1.zdclitem01;  obj\_gmhipf.zdclitem02 := obj\_mbrindp1.zdclitem02;  obj\_gmhipf.zdcrsncd := o\_defaultvalues('ZDCRSNCD');  obj\_gmhipf.zdeclcat := c\_space;  obj\_gmhipf.zdfcncy := obj\_mbrindp1.zdfcncy;  obj\_gmhipf.datatype := o\_defaultvalues('DATATYPE');  /\* select SUM(APREM)  into v\_zprmsi  from STAGEDBUSR.TITDMGMBRINDP2  where TRIM(REFNUM) = TRIM(obj\_mbrindp1.refnum); \*/  obj\_gmhipf.zprmsi := v\_zprmsi;  obj\_gmhipf.zadchctl := o\_defaultvalues('ZADCHCTL');  obj\_gmhipf.zmargnflg := obj\_mbrindp1.zmargnflg;  obj\_gmhipf.user\_t := i\_vrcmuser;  -- obj\_gmhipf.ZALTRCDE01 := o\_defaultvalues('ZALTRCDE01');  -- obj\_gmhipf.ZALTRCDE02 := o\_defaultvalues('ZALTRCDE02');  -- obj\_gmhipf.ZALTRCDE03 := o\_defaultvalues('ZALTRCDE03');  -- obj\_gmhipf.ZALTRCDE04 := o\_defaultvalues('ZALTRCDE04');  -- obj\_gmhipf.ZALTRCDE05 := o\_defaultvalues('ZALTRCDE05');  obj\_gmhipf.dcldate := obj\_mbrindp1.dcldate;  obj\_gmhipf.notsfrom := n\_issdate;  obj\_gmhipf.zstatresn := obj\_mbrindp1.zstatresn;  obj\_gmhipf.hpropdte := obj\_mbrindp1.hpropdte;  -- obj\_gmhipf.ADDRINDC := o\_defaultvalues('ADDRINDC');  obj\_gmhipf.addrindc := c\_space; --MB7  obj\_gmhipf.zwrkpct := o\_defaultvalues('ZWRKPCT');  obj\_gmhipf.zinhdsclm := o\_defaultvalues('ZINHDSCLM');  obj\_gmhipf.docrcdte := obj\_mbrindp1.docrcvdt;  -- obj\_gmhipf.ISSTAFF := C\_SPACE; --MB7 comment out as setting null  obj\_gmhipf.zplancde := obj\_mbrindp1.zplancde;  obj\_gmhipf.zintent := o\_defaultvalues('ZINTENT');  ---SIT Changes  obj\_gmhipf.zdeclcat := obj\_mbrindp1.zdeclcat;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO GMHIPF VALUES obj\_gmhipf;  ---MB18  gmhipfindex := gmhipfindex + 1;  gmhipf\_list.extend;  gmhipf\_list(gmhipfindex) := obj\_gmhipf;  ---MB18  --------insert into IG table "GMOVPF" for GMHIPF BEGIN(pq9ho.updateGmhdpf())----  --IF (TRIM(obj\_mbrindp1.statcode) <> 'PN') THEN Previous code  IF ( trim(obj\_mbrindp1.statcode) = 'XN' ) THEN  v\_olddta2 := obj\_gmhipf.chdrcoy  || obj\_gmhipf.chdrnum  || obj\_gmhipf.mbrno  || obj\_gmhipf.dpntno  || obj\_gmhipf.effdate  || c\_maxdate  || obj\_gmhipf.subscoy  || obj\_gmhipf.subsnum  || obj\_gmhipf.occpcode  || obj\_gmhipf.salary  || obj\_gmhipf.dteapp  || obj\_gmhipf.sbstdl  || obj\_gmhipf.termid  || obj\_gmhipf.user\_t  || obj\_gmhipf.trdt  || obj\_gmhipf.trtm  || obj\_gmhipf.tranno  || obj\_gmhipf.fupflg  || obj\_gmhipf.dpntno  || obj\_gmhipf.client  || obj\_gmhipf.personcov  || obj\_gmhipf.mlvlplan  || obj\_gmhipf.clntcoy  || obj\_gmhipf.pccclnt  || obj\_gmhipf.apccclnt  || obj\_gmhipf.earning  || obj\_gmhipf.ctbprcnt  || obj\_gmhipf.ctbamt  || obj\_gmhipf.usrprf  || obj\_gmhipf.jobnm  || obj\_gmhipf.datime  || obj\_gmhipf.isstaff  || obj\_gmhipf.ztrxstat  || obj\_gmhipf.zstatresn  || obj\_gmhipf.quoteno  || obj\_gmhipf.zplancde  || obj\_gmhipf.hpropdte  || obj\_gmhipf.notsfrom  || obj\_gmhipf.docrcdte  || obj\_gmhipf.dcldate  || obj\_gmhipf.zdeclcat  || obj\_gmhipf.zdclitem01  || obj\_gmhipf.zdclitem02  || obj\_gmhipf.zdcrsncd  ||  -- obj\_gmhipf.Zaltrcde01 ||  -- obj\_gmhipf.Zaltrcde02 || obj\_gmhipf.Zaltrcde03 ||  -- obj\_gmhipf.Zaltrcde04 || obj\_gmhipf.Zaltrcde05 ||  obj\_gmhipf.addrindc  || obj\_gmhipf.zadchctl  || obj\_gmhipf.zmargnflg  || obj\_gmhipf.zdfcncy  || obj\_gmhipf.zinhdsclm  || obj\_gmhipf.zcpnscde  || obj\_gmhipf.zprmsi;  v\_refkey := obj\_gmhipf.chdrcoy  || obj\_gmhipf.chdrnum  || obj\_gmhipf.mbrno  || obj\_gmhipf.dpntno  || obj\_gmhipf.effdate;  obj\_gmovpf2.chdrcoy := obj\_gmhipf.chdrcoy;  obj\_gmovpf2.chdrnum := obj\_gmhipf.chdrnum;  obj\_gmovpf2.effdate := obj\_mbrindp1.effdate;  obj\_gmovpf2.tranno := 1;  obj\_gmovpf2.user\_t := obj\_gmhipf.user\_t;  obj\_gmovpf2.trdt := obj\_gmhipf.trdt;  obj\_gmovpf2.trtm := obj\_gmhipf.trtm;  obj\_gmovpf2.termid := obj\_gmhipf.termid;  obj\_gmovpf2.batccoy := i\_company;  obj\_gmovpf2.batcbrn := i\_branch;  obj\_gmovpf2.batcactyr := i\_acctyear;  obj\_gmovpf2.batcactmn := i\_acctmonth;  IF ( trim(obj\_mbrindp1.statcode) = 'CA' ) THEN  obj\_gmovpf2.batctrcde := 'T913';  ELSE  obj\_gmovpf2.batctrcde := 'T903';  END IF;  obj\_gmovpf2.batcbatch := c\_space;  obj\_gmovpf2.olddta := v\_olddta2;  obj\_gmovpf2.newdta := '';  obj\_gmovpf2.funccode := c\_funcdeaddmbr;  obj\_gmovpf2.refkey := v\_refkey;  obj\_gmovpf2.rfmt := 'GMHIREC';  obj\_gmovpf2.chgtype := '';  obj\_gmovpf2.riprior := ' ';  obj\_gmovpf2.onpreflg := '';  obj\_gmovpf2.fachold := '';  obj\_gmovpf2.planvflg := '';  obj\_gmovpf2.usrprf := i\_usrprf;  obj\_gmovpf2.jobnm := i\_schedulename;  obj\_gmovpf2.datime := CAST ( SYSDATE AS TIMESTAMP );  SELECT  seq\_gmovpf.NEXTVAL  INTO v\_seq\_gmovpf  FROM  dual;  obj\_gmovpf2.unique\_number := v\_seq\_gmovpf;  --INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO GMOVPF VALUES obj\_gmovpf2;  ---MB18  gmovpf2index := gmovpf2index + 1;  gmovpf2\_list.extend;  gmovpf2\_list(gmovpf2index) := obj\_gmovpf2;  ---MB18  END IF;  /\* IF (TRIM(obj\_mbrindp1.statcode) = 'CA') THEN  v\_olddta2 := obj\_gmhipf.Chdrcoy || obj\_gmhipf.Chdrnum ||  obj\_gmhipf.Mbrno || obj\_gmhipf.Dpntno ||  obj\_gmhipf.Effdate || C\_MAXDATE || obj\_gmhipf.Subscoy ||  obj\_gmhipf.Subsnum || obj\_gmhipf.Occpcode ||  obj\_gmhipf.Salary || obj\_gmhipf.Dteapp ||  obj\_gmhipf.Sbstdl || obj\_gmhipf.Termid ||  obj\_gmhipf.User\_t || obj\_gmhipf.Trdt ||  obj\_gmhipf.Trtm || obj\_gmhipf.Tranno ||  obj\_gmhipf.Fupflg || obj\_gmhipf.Dpntno ||  obj\_gmhipf.Client || obj\_gmhipf.Personcov ||  obj\_gmhipf.Mlvlplan || obj\_gmhipf.Clntcoy ||  obj\_gmhipf.Pccclnt || obj\_gmhipf.Apccclnt ||  obj\_gmhipf.Earning || obj\_gmhipf.Ctbprcnt ||  obj\_gmhipf.Ctbamt || obj\_gmhipf.Usrprf ||  obj\_gmhipf.Jobnm || obj\_gmhipf.Datime ||  obj\_gmhipf.Isstaff || obj\_gmhipf.Ztrxstat ||  obj\_gmhipf.Zstatresn || obj\_gmhipf.Quoteno ||  obj\_gmhipf.Zplancde || obj\_gmhipf.Hpropdte ||  obj\_gmhipf.Notsfrom || obj\_gmhipf.Docrcdte ||  obj\_gmhipf.Dcldate || obj\_gmhipf.Zdeclcat ||  obj\_gmhipf.Zdclitem01 || obj\_gmhipf.Zdclitem02 ||  obj\_gmhipf.Zdcrsncd ||  --obj\_gmhipf.Zaltrcde01 ||  -- obj\_gmhipf.Zaltrcde02 || obj\_gmhipf.Zaltrcde03 ||  -- obj\_gmhipf.Zaltrcde04 || obj\_gmhipf.Zaltrcde05 ||  obj\_gmhipf.Addrindc || obj\_gmhipf.Zadchctl ||  obj\_gmhipf.Zmargnflg || obj\_gmhipf.Zdfcncy ||  obj\_gmhipf.Zinhdsclm || obj\_gmhipf.Zcpnscde ||  obj\_gmhipf.Zprmsi;  v\_newdta2 := obj\_gmhipf.Chdrcoy || obj\_gmhipf.Chdrnum ||  obj\_gmhipf.Mbrno || obj\_gmhipf.Dpntno ||  obj\_gmhipf.Effdate || obj\_gmhipf.Dtetrm ||  obj\_gmhipf.Subscoy || obj\_gmhipf.Subsnum ||  obj\_gmhipf.Occpcode || obj\_gmhipf.Salary ||  obj\_gmhipf.Dteapp || obj\_gmhipf.Sbstdl ||  obj\_gmhipf.Termid || obj\_gmhipf.User\_t ||  obj\_gmhipf.Trdt || obj\_gmhipf.Trtm ||  obj\_gmhipf.Tranno || obj\_gmhipf.Fupflg ||  obj\_gmhipf.Dpntno || obj\_gmhipf.Client ||  obj\_gmhipf.Personcov || obj\_gmhipf.Mlvlplan ||  obj\_gmhipf.Clntcoy || obj\_gmhipf.Pccclnt ||  obj\_gmhipf.Apccclnt || obj\_gmhipf.Earning ||  obj\_gmhipf.Ctbprcnt || obj\_gmhipf.Ctbamt ||  obj\_gmhipf.Usrprf || obj\_gmhipf.Jobnm ||  obj\_gmhipf.Datime || obj\_gmhipf.Isstaff ||  obj\_gmhipf.Ztrxstat || obj\_gmhipf.Zstatresn ||  obj\_gmhipf.Quoteno || obj\_gmhipf.Zplancde ||  obj\_gmhipf.Hpropdte || obj\_gmhipf.Notsfrom ||  obj\_gmhipf.Docrcdte || obj\_gmhipf.Dcldate ||  obj\_gmhipf.Zdeclcat || obj\_gmhipf.Zdclitem01 ||  obj\_gmhipf.Zdclitem02 || obj\_gmhipf.Zdcrsncd ||  -- obj\_gmhipf.Zaltrcde01 || obj\_gmhipf.Zaltrcde02 ||  -- obj\_gmhipf.Zaltrcde03 || obj\_gmhipf.Zaltrcde04 ||  -- obj\_gmhipf.Zaltrcde05 ||  obj\_gmhipf.Addrindc || obj\_gmhipf.Zadchctl ||  obj\_gmhipf.Zmargnflg || obj\_gmhipf.Zdfcncy ||  obj\_gmhipf.Zinhdsclm || obj\_gmhipf.Zcpnscde ||  obj\_gmhipf.Zprmsi;  v\_refkey := obj\_gmhipf.Chdrcoy + obj\_gmhipf.Chdrnum +  obj\_gmhipf.Mbrno + obj\_gmhipf.Dpntno +  obj\_gmhipf.Effdate;  obj\_gmovpf2.Chdrcoy := obj\_gmhipf.Chdrcoy;  obj\_gmovpf2.Chdrnum := obj\_gmhipf.Chdrnum;  obj\_gmovpf2.Effdate := obj\_gmhipf.effdate;  obj\_gmovpf2.Tranno := 2;  obj\_gmovpf2.User\_t := obj\_gmhipf.User\_t;  obj\_gmovpf2.Trdt := obj\_gmhipf.Trdt;  obj\_gmovpf2.Trtm := obj\_gmhipf.Trtm;  obj\_gmovpf2.Termid := obj\_gmhipf.Termid;  obj\_gmovpf2.Batccoy := i\_company;  obj\_gmovpf2.Batcbrn := i\_branch;  obj\_gmovpf2.Batcactyr := i\_acctYear;  obj\_gmovpf2.Batcactmn := i\_acctMonth;  IF (TRIM(obj\_mbrindp1.statcode) = 'CA') THEN  obj\_gmovpf2.Batctrcde := 'T913';  ELSE  obj\_gmovpf2.Batctrcde := 'T903';  END IF;  obj\_gmovpf2.Batcbatch := C\_SPACE;  obj\_gmovpf2.Olddta := v\_olddta2;  obj\_gmovpf2.Newdta := v\_newdta2;  obj\_gmovpf2.Funccode := C\_FUNCDETRMBR;  obj\_gmovpf2.Refkey := v\_refkey;  obj\_gmovpf2.Rfmt := 'GMHIREC';  obj\_gmovpf2.Chgtype := '';  obj\_gmovpf2.Riprior := ' ';  obj\_gmovpf2.Onpreflg := '';  obj\_gmovpf2.Fachold := '';  obj\_gmovpf2.Planvflg := '';  obj\_gmovpf2.Usrprf := i\_usrprf;  obj\_gmovpf2.Jobnm := i\_scheduleName;  obj\_gmovpf2.Datime := CAST(sysdate AS TIMESTAMP);  Insert into GMOVPF values obj\_gmovpf2;  END IF;\*/  --------insert into IG table "GMOVPF" for GMHIPF END (pq9ho.updateGmhdpf())----  ---Insert into IG table "GMHIPF" END ---  /\* ITR3 Changes removed ZMCIPF table  ---Insert into IG table "ZMCIPF " BEGIN (ZmcipfDAOImpl.updateZmcipf(), pq9ho.updateUniquePerEndorserTab) ---  -- obj\_zmcipf.CHDRCOY := 'a'; --Not present in Table,Not even mentioned in Data model  obj\_zmcipf.CHDRNUM := SUBSTR(TRIM(obj\_mbrindp1.refnum), 1, 8);  --obj\_zmcipf.MBRNO := 'a'; --Not present in Table,Not even mentioned in Data model  IF (TRIM(obj\_mbrindp1.statcode) = 'CA') THEN  obj\_zmcipf.TRANNO := 2;  ELSE  obj\_zmcipf.TRANNO := 1;  END IF;  obj\_zmcipf.ZENDCDE := obj\_mbrindp1.zendcde; -- obj\_zmcipf.ZENDCODE := 'a';  obj\_zmcipf.ZENSPCD01 := obj\_mbrindp1.zenspcd01;  IF (((TRIM(obj\_mbrindp1.zenspcd02) IS NULL) OR  ((TRIM(obj\_mbrindp1.zcifcode))) IS NULL) AND  ((TRIM(obj\_mbrindp1.zenspcd01) IS NOT NULL))) THEN  select ZENSPCD02, ZCIFCODE  into v\_zenspcd02\_1, v\_zcifcode  from ZCLEPF  where TRIM(clntnum) = TRIM(v\_zigvalue)  and TRIM(zenspcd01) = TRIM(obj\_mbrindp1.zenspcd01)  and TRIM(zendcde) = TRIM(obj\_mbrindp1.zendcde);  END IF;  IF ((TRIM(obj\_mbrindp1.zenspcd02)) IS NOT NULL) THEN  obj\_zmcipf.ZENSPCD02 := obj\_mbrindp1.zenspcd02;  ELSE  obj\_zmcipf.ZENSPCD02 := v\_zenspcd02\_1;  END IF;  IF ((TRIM(obj\_mbrindp1.zcifcode)) IS NOT NULL) THEN  obj\_zmcipf.ZCIFCODE := obj\_mbrindp1.zcifcode;  ELSE  obj\_zmcipf.ZCIFCODE := v\_zcifcode;  END IF;  obj\_zmcipf.CRDTCARD := obj\_mbrindp1.crdtcard;  obj\_zmcipf.BANKACCKEY01 := obj\_mbrindp1.bnkacckey01;  \\* BEGIN  IF (TRIM(obj\_mbrindp1.bnkacckey01) IS NOT NULL) THEN  select bankaccdsc, bnkactyp, bankkey  into v\_bankaccdsc, v\_bnkactyp, v\_bankkey  from CLBAPF  where TRIM(bankacckey) = TRIM(obj\_mbrindp1.bnkacckey01)  AND TRIM(Clntpfx) = TRIM('CN')  and TRIM(Clntcoy) = TRIM(i\_fsucocompany)  and TRIM(clntnum) = TRIM(v\_zigvalue)  and TRIM(validflag) = TRIM('1')  and TRIM(bnkactyp) = TRIM('CC');  END IF;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  v\_bankaccdsc := null;  v\_bnkactyp := null;  v\_bankkey := null;  END;  \*\  IF (TRIM(obj\_mbrindp1.bnkacckey01) IS NOT NULL) THEN  IF (getclba.exists(TRIM(obj\_mbrindp1.bnkacckey01) ||  TRIM(v\_zigvalue) || TRIM(i\_fsucocompany))) THEN  v\_obj\_clba := getclba(TRIM(obj\_mbrindp1.bnkacckey01) ||  TRIM(v\_zigvalue) || TRIM(i\_fsucocompany));  v\_bankaccdsc := v\_obj\_clba.bankaccdsc;  v\_bnkactyp := v\_obj\_clba.bnkactyp;  v\_bankkey := v\_obj\_clba.bankkey;  END IF;  END IF;  obj\_zmcipf.BANKACCDSC01 := v\_bankaccdsc;  obj\_zmcipf.BNKACTYP01 := v\_bnkactyp;  obj\_zmcipf.BANKACCKEY02 := C\_SPACE;  obj\_zmcipf.BANKACCDSC02 := C\_SPACE;  obj\_zmcipf.BNKACTYP02 := C\_SPACE;  -- obj\_zmcipf.ZBANKCD := 'a';  -- obj\_zmcipf.ZBRANCHCD := 'a';  obj\_zmcipf.ZPBCTYPE := C\_SPACE;  obj\_zmcipf.ZPBCODE := C\_SPACE;  obj\_zmcipf.PREAUTNO := obj\_mbrindp1.preautno;  \\* BEGIN  IF (TRIM(obj\_mbrindp1.crdtcard) IS NOT NULL) THEN  select MTHTO, YEARTO  into v\_mthto, v\_yearto  from CLBAPF  where TRIM(bankacckey) = TRIM(obj\_mbrindp1.crdtcard)  AND TRIM(Clntpfx) = TRIM('CN')  and TRIM(Clntcoy) = TRIM(i\_fsucocompany)  and TRIM(clntnum) = TRIM(v\_zigvalue)  and TRIM(validflag) = TRIM('1')  and TRIM(bnkactyp) = TRIM('CC');  END IF;  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  obj\_zmcipf.MTHTO := 0;  obj\_zmcipf.YEARTO := 0;  END;\*\  IF (TRIM(obj\_mbrindp1.crdtcard) IS NOT NULL) THEN  IF (getclba.exists(TRIM(obj\_mbrindp1.crdtcard) || TRIM(v\_zigvalue) ||  TRIM(i\_fsucocompany))) THEN  v\_obj\_clba := getclba(TRIM(obj\_mbrindp1.crdtcard) ||  TRIM(v\_zigvalue) || TRIM(i\_fsucocompany));  v\_mthto := v\_obj\_clba.mthto;  v\_yearto := v\_obj\_clba.yearto;  END IF;  END IF;  obj\_zmcipf.MTHTO := v\_mthto;  obj\_zmcipf.YEARTO := v\_yearto;  obj\_zmcipf.DATIME := CAST(sysdate AS TIMESTAMP);  obj\_zmcipf.JOBNM := i\_scheduleName;  BEGIN  IF (TRIM(obj\_mbrindp1.mpolnum) IS NOT NULL) THEN  IF LENGTH(TRIM(obj\_mbrindp1.crdtcard)) > 6 THEN  v\_temp\_crdtcard := SUBSTR(TRIM(obj\_mbrindp1.crdtcard), 0, 6);  -- dbms\_output.put\_line('v\_temp\_crdtcard--'||v\_temp\_crdtcard);  END IF;  SELECT ZCRDTYPE  into v\_zcrdtype  FROM ZENCTPF  WHERE TRIM(ZPOLNMBR) = obj\_mbrindp1.mpolnum  and ((TRIM(TO\_NUMBER(ZCNBRFRM)) < TRIM(v\_temp\_crdtcard) and  TRIM(TO\_NUMBER(ZCNBRTO)) > TRIM(v\_temp\_crdtcard)) OR  TRIM(TO\_NUMBER(ZCNBRFRM)) = TRIM(v\_temp\_crdtcard) OR  TRIM(TO\_NUMBER(ZCNBRTO)) = TRIM(v\_temp\_crdtcard));  ELSE  SELECT ZCRDTYPE  into v\_zcrdtype  FROM ZENCTPF  WHERE TRIM(ZENDCDE) = TRIM(obj\_mbrindp1.zendcde)  and ((TRIM(TO\_NUMBER(ZCNBRFRM)) < TRIM(v\_temp\_crdtcard) and  TRIM(TO\_NUMBER(ZCNBRTO)) > TRIM(v\_temp\_crdtcard)) OR  TRIM(TO\_NUMBER(ZCNBRFRM)) = TRIM(v\_temp\_crdtcard) OR  TRIM(TO\_NUMBER(ZCNBRTO)) = TRIM(v\_temp\_crdtcard));  END IF;  EXCEPTION  when No\_data\_found then  v\_zcrdtype := null;  end;  obj\_zmcipf.CARDTYP := v\_zcrdtype;  obj\_zmcipf.BANKKEY := v\_bankkey;  obj\_zmcipf.USRPRF := i\_usrprf;  -- obj\_zmcipf.DPNTNO := 'a'; --Not present in Table,Not even mentioned in Data model  insert into ZMCIPF values obj\_zmcipf;  ---Insert into IG table "ZMCIPF" END ---\*/  ---Insert into IG table "ZCLEPF" START ---\*/  --- 17/02/2018 After Pre-SIT execution  --- IF (TRIM(obj\_mbrindp1.ZENSPCD01) IS NOT NULL) THEN  IF ( ( TRIM(obj\_mbrindp1.zenspcd01) IS NOT NULL ) OR ( TRIM(obj\_mbrindp1.zenspcd02) IS NOT NULL ) OR ( TRIM(obj\_mbrindp1  .zcifcode) IS NOT NULL ) ) THEN  -- MB12 : MOD : Cratte ZCLEPF iff one of them given  SELECT  seq\_zclepf.NEXTVAL  INTO v\_seq\_zclepf  FROM  dual;  obj\_zclepf.unique\_number := v\_seq\_zclepf;  obj\_zclepf.clntnum := obj\_gchd.cownnum;  obj\_zclepf.zendcde := obj\_gchppf.zendcde;  obj\_zclepf.zenspcd01 := obj\_mbrindp1.zenspcd01;  obj\_zclepf.zenspcd02 := obj\_mbrindp1.zenspcd02;  obj\_zclepf.zcifcode := obj\_mbrindp1.zcifcode;  obj\_zclepf.usrprf := i\_usrprf; --MB7 New Added  obj\_zclepf.jobnm := i\_schedulename; --MB7 New Added  obj\_zclepf.datime := CAST ( SYSDATE AS TIMESTAMP ); --MB7 New Added  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO ZCLEPF VALUES obj\_zclepf;  ---MB18  zclepfindex := zclepfindex + 1;  zclepf\_list.extend;  zclepf\_list(zclepfindex) := obj\_zclepf;  ---MB18  ---Insert into IG table "ZCLEPF" END ---\*/  END IF;  ---Insert into IG table "GPSUPF" END ---\*/  SELECT  seq\_gpsupf.NEXTVAL  INTO v\_seq\_gpsupf  FROM  dual;  obj\_gpsupf.unique\_number := v\_seq\_gpsupf;  obj\_gpsupf.chdrcoy := i\_company;  obj\_gpsupf.chdrnum := substr(trim(obj\_mbrindp1.refnum), 1, 8);  obj\_gpsupf.subscoy := i\_fsucocompany;  /\* select ZCLNTID  into v\_zclntid  from ZENDRPF  where TRIM(ZENDCDE) = TRIM(obj\_gchppf.ZENDCDE);\*/  IF ( zendcde.EXISTS(trim(obj\_mbrindp1.zendcde)) ) THEN  v\_zclntid := zendcde(trim(obj\_mbrindp1.zendcde));  ELSE  v\_zclntid := NULL;  END IF;  obj\_gpsupf.subsnum := v\_zclntid;  obj\_gpsupf.dteatt := obj\_gchipf.ccdate;  obj\_gpsupf.dtetrm := c\_maxdate;  obj\_gpsupf.reasontrm := c\_space;  obj\_gpsupf.lnbillno := c\_zero;  obj\_gpsupf.labillno := c\_zero;  obj\_gpsupf.lpbillno := c\_zero;  obj\_gpsupf.ptdate := c\_maxdate;  obj\_gpsupf.ptdateab := c\_maxdate;  obj\_gpsupf.termid := i\_vrcmtermid;  obj\_gpsupf.user\_t := i\_vrcmuser;  obj\_gpsupf.trdt := i\_trdt;  obj\_gpsupf.trtm := i\_vrcmtime;  obj\_gpsupf.tranno := '1';  obj\_gpsupf.schdflg := c\_space;  obj\_gpsupf.mandref := c\_space;  obj\_gpsupf.usrprf := i\_usrprf;  obj\_gpsupf.jobnm := i\_schedulename;  obj\_gpsupf.datime := CAST ( SYSDATE AS TIMESTAMP );  obj\_gpsupf.sinfdte := c\_maxdate;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO gpsupf VALUES obj\_gpsupf;  ---MB18  gpsupfindex := gpsupfindex + 1;  gpsupf\_list.extend;  gpsupf\_list(gpsupfindex) := obj\_gpsupf;  ---MB18  ---Insert into IG table "GPSUPF" END ---\*/  ---ITR4 CHANGE INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ ZCELINKPF START----------------  obj\_zcelinkpf.clntpfx := trim(o\_defaultvalues('CLNTPFX'));  obj\_zcelinkpf.clntcoy := trim(i\_fsucocompany);  obj\_zcelinkpf.clntnum := trim(obj\_gchd.cownnum);  obj\_zcelinkpf.zendcde := trim(obj\_mbrindp1.zendcde);  obj\_zcelinkpf.usrprf := i\_usrprf;  obj\_zcelinkpf.jobnm := i\_schedulename;  obj\_zcelinkpf.datime := CAST ( SYSDATE AS TIMESTAMP );  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO VM1DTA.VIEW\_DM\_ZCELINKPF VALUES obj\_zcelinkpf;  ---MB18  zcelinkpfindex := zcelinkpfindex + 1;  zcelinkpf\_list.extend;  zcelinkpf\_list(zcelinkpfindex) := obj\_zcelinkpf;  ---MB18  ---ITR4 CHANGE INSERT ZCELINKPF END----------------  INSERT INTO zdrppf (  chdrnum,  prefix,  jobnum,  jobname  ) VALUES (  TRIM(v\_refnump1),  v\_prefix,  i\_schedulenumber,  i\_schedulename  );  END IF;  /\* mbrmigindex := mbrmigindex + 1;  mbrmig\_list.extend;  mbrmig\_list(mbrmigindex) := obj\_mbrmig;\*/  END LOOP;  idx := gchd\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gchd\_list.first..gchd\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gchd VALUES gchd\_list ( idx );  END IF;  idx := gchppf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gchppf\_list.first..gchppf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gchppf VALUES gchppf\_list ( idx );  END IF;  idx := gchipf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gchipf\_list.first..gchipf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gchipf VALUES gchipf\_list ( idx );  END IF;  idx := gmovpf1\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gmovpf1\_list.first..gmovpf1\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gmovpf VALUES gmovpf1\_list ( idx );  END IF;  idx := gmhipf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gmhipf\_list.first..gmhipf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gmhipf VALUES gmhipf\_list ( idx );  END IF;  idx := gmhdpf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gmhdpf\_list.first..gmhdpf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gmhdpf VALUES gmhdpf\_list ( idx );  END IF;  idx := gmovpf2\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gmovpf2\_list.first..gmovpf2\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gmovpf VALUES gmovpf2\_list ( idx );  END IF;  idx := zclepf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN zclepf\_list.first..zclepf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.zclepf VALUES zclepf\_list ( idx );  END IF;  idx := gpsupf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gpsupf\_list.first..gpsupf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gpsupf VALUES gpsupf\_list ( idx );  END IF;  idx := zcelinkpf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN zcelinkpf\_list.first..zcelinkpf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.view\_dm\_zcelinkpf VALUES zcelinkpf\_list ( idx );  END IF;  EXIT WHEN cur\_mbr\_ind\_p1%notfound;  gchd\_list.DELETE;  gchppf\_list.DELETE;  gchipf\_list.DELETE;  gmovpf1\_list.DELETE;  gmhipf\_list.DELETE;  gmhdpf\_list.DELETE;  gmovpf2\_list.DELETE;  zclepf\_list.DELETE;  gpsupf\_list.DELETE;  zcelinkpf\_list.DELETE;  gchdindex := 0;  gchppfindex := 0;  gchipfindex := 0;  gmovpf1index := 0;  gmhipfindex := 0;  gmhdpfindex := 0;  gmovpf2index := 0;  zclepfindex := 0;  gpsupfindex := 0;  zcelinkpfindex := 0;  END LOOP;  CLOSE cur\_mbr\_ind\_p1;  /\* OPEN cur\_mbr\_ind\_p2;  <<skipRecordp2>>  LOOP  FETCH cur\_mbr\_ind\_p2  INTO obj\_mbrindp2;  EXIT WHEN cur\_mbr\_ind\_p2%NOTFOUND;\*/  OPEN cur\_mbr\_ind\_p2;  LOOP  FETCH cur\_mbr\_ind\_p2 BULK COLLECT INTO mbrind2\_list LIMIT 5000;  << skiprecordp2 >> FOR i IN 1..mbrind2\_list.count LOOP  -- dbms\_output.put\_line('Loop 2 start');  obj\_mbrindp2 := mbrind2\_list(i);  v\_refkeyp2 := trim(obj\_mbrindp2.refnum)  || '-'  || trim(obj\_mbrindp2.prodtyp);  -- dbms\_output.put\_line('v\_refKeyp2 ==>' || v\_refKeyp2);  -- v\_temprefKeyp2 := '''' || v\_refKeyp2 || '''';  -- v\_constructrefkey := CONCAT(v\_constructrefkey, v\_temprefKeyp2);  -- v\_constructrefkey := CONCAT(v\_constructrefkey, ',');  ----------MB13: START-----------------  v\_seqno1 := substr(trim(obj\_mbrindp2.refnum), 9, 3);  v\_seqnoincr := v\_seqno1 + 1;  v\_seqno := 1;  ----------MB13: ENDS---------------------  ----------Initialization -------  ---For PGA  ---MB18:  i\_refnum := obj\_mbrindp2.refnum\_c;  i\_cnttypind := obj\_mbrindp2.cnttypind\_c;  i\_mpolnum := obj\_mbrindp2.mpolnum\_c;  i\_statcode := obj\_mbrindp2.statcode\_c;  i\_dtetrm := obj\_mbrindp2.dtetrm\_c;  i\_zplancde := obj\_mbrindp2.zplancde\_c;  i\_effdate := obj\_mbrindp2.zwaitpedt\_c;  i\_zwaitpedt := obj\_mbrindp2.effdate\_c;  i\_zpoltdate := obj\_mbrindp2.zpoltdate\_c;  i\_zpdatatxflag := obj\_mbrindp2.zpdatatxflg\_c;  i\_ztrxstat := obj\_mbrindp2.ztrxstat\_c;  ---MB18:  /\*IF (mbrp1info.exists(TRIM(obj\_mbrindp2.refnum))) THEN  i\_refnum := mbrp1info(TRIM(obj\_mbrindp2.refnum)).refnum;  i\_cnttypind := mbrp1info(TRIM(obj\_mbrindp2.refnum)).cnttypind;  i\_mpolnum := mbrp1info(TRIM(obj\_mbrindp2.refnum)).mpolnum;  i\_statcode := mbrp1info(TRIM(obj\_mbrindp2.refnum)).statcode;  i\_dtetrm := mbrp1info(TRIM(obj\_mbrindp2.refnum)).dtetrm;  i\_zplancde := mbrp1info(TRIM(obj\_mbrindp2.refnum)).zplancde;  i\_effdate := mbrp1info(TRIM(obj\_mbrindp2.refnum)).effdate;  i\_ZWAITPEDT := mbrp1info(TRIM(obj\_mbrindp2.refnum)).zwaitpedt;  i\_zpoltdate := mbrp1info(TRIM(obj\_mbrindp2.refnum)).zpoltdate;  i\_zpdatatxflag := mbrp1info(TRIM(obj\_mbrindp2.refnum)).zpdatatxflag;  i\_ztrxstat := mbrp1info(TRIM(obj\_mbrindp2.refnum)).ztrxstat;  --MB16 IF (b\_checkpolst.exists(TRIM(i\_refnum))) THEN  --MB16 i\_statcode := b\_checkpolst(i\_refnum);  --MB16 END IF; ---MB12  -- dbms\_output.put\_line('i\_refnum = ' || i\_refnum || 'i\_cnttypind=' || i\_cnttypind || 'i\_mpolnum=' || i\_mpolnum || 'i\_statcode=' || i\_statcode || 'i\_dtetrm=' || i\_dtetrm || 'i\_zplancde=' || i\_zplancde || 'i\_effdate=' || i\_effdate);  --FOR PGA  /\*  ELSE  CONTINUE skipRecordp2;  END IF;\*/  IF trim(i\_cnttypind) = 'I' THEN  v\_prefix := c\_prefix\_indv;  ELSE  v\_prefix := c\_prefix\_mebr;  END IF;  i\_zdoe\_infop2 := NULL;  i\_zdoe\_infop2.i\_zfilename := 'TITDMGMBRINDP2';  i\_zdoe\_infop2.i\_prefix := v\_prefix;  i\_zdoe\_infop2.i\_scheduleno := i\_schedulenumber;  i\_zdoe\_infop2.i\_refkey := trim(v\_refkeyp2);  IF ( trim(v\_prefix) = 'MB' ) THEN  i\_zdoe\_infop2.i\_tablename := v\_tablenamemb;  ELSE  i\_zdoe\_infop2.i\_tablename := v\_tablenamein;  END IF;  /\* IF (TRIM(v\_prefix) = 'MB') THEN  -- i\_zdoe\_infop2.i\_tablecnt := v\_zdoecrtcountMB;  v\_zdoecrtcountMB := 1;  ELSE  i\_zdoe\_infop2.i\_tablecnt := v\_zdoecrtcountIN;  v\_zdoecrtcountIN := 1;  END IF;\*/  v\_isanyerrorp2 := 'N';  v\_errorcountp2 := 0;  t\_ercodep2(1) := NULL;  t\_ercodep2(2) := NULL;  t\_ercodep2(3) := NULL;  t\_ercodep2(4) := NULL;  t\_ercodep2(5) := NULL;  /\*  gxhipfindex := 0;  gmovpf3PNindex := 0;  gmovpf3CAindex := 0;  gaphpfindex := 0;  mtrnpfCAindex := 0;  mtrnpfPNindex := 0;\*/  v\_zprmsi := v\_zprmsi + obj\_mbrindp2.aprem;  ----------Initialization -------  ---------------First part of validation -TITDMGMBRINDP2----------------------------------------  ---REFNUM NULL VALIDATION  IF TRIM(obj\_mbrindp2.refnum) IS NULL THEN  v\_isanyerrorp2 := 'Y';  v\_globalerrorp2 := 'Y';  v\_errorcountp2 := v\_errorcountp2 + 1;  t\_ercodep2(v\_errorcountp2) := c\_z082;  t\_errorfieldp2(v\_errorcountp2) := 'REFNUM';  t\_errormsgp2(v\_errorcountp2) := o\_errortext(c\_z082);  t\_errorfieldvalp2(v\_errorcountp2) := obj\_mbrindp2.refnum;  t\_errorprogramp2(v\_errorcountp2) := i\_schedulename;  GOTO insertzdoep2;  END IF;  ---PRODTYP NULL VALIDATION  IF TRIM(obj\_mbrindp2.prodtyp) IS NULL THEN  v\_isanyerrorp2 := 'Y';  v\_globalerrorp2 := 'Y';  v\_errorcountp2 := v\_errorcountp2 + 1;  t\_ercodep2(v\_errorcountp2) := c\_z091;  t\_errorfieldp2(v\_errorcountp2) := 'PRODTYP';  t\_errormsgp2(v\_errorcountp2) := o\_errortext(c\_z091);  t\_errorfieldvalp2(v\_errorcountp2) := obj\_mbrindp2.prodtyp;  t\_errorprogramp2(v\_errorcountp2) := i\_schedulename;  GOTO insertzdoep2;  END IF;  /\* MPS Performance Issue - There is no need to perform this as PK should not allow duplicates  --REFNUM+PRODTYP duplicate check validation  IF ((TRIM(obj\_mbrindp2.prodtyp) IS NOT NULL) AND  (TRIM(obj\_mbrindp2.refnum) IS NOT NULL)) THEN  select count(\*)  into v\_isduplicatep2  from TITDMGMBRINDP2@DATABASE\_LINK1  where TRIM(REFNUM) = TRIM(obj\_mbrindp2.refnum)  AND TRIM(PRODTYP) = TRIM(obj\_mbrindp2.prodtyp);  IF TRIM(v\_isduplicatep2) > 1 THEN  v\_GlobalErrorp2 := 'Y';  v\_errorCountp2 := v\_errorCountp2 + 1;  t\_ercodep2(v\_errorCountp2) := C\_Z099;  t\_errorfieldp2(v\_errorCountp2) := 'REFNUM';  t\_errormsgp2(v\_errorCountp2) := o\_errortext(C\_Z099);  t\_errorfieldvalp2(v\_errorCountp2) := obj\_mbrindp2.refnum;  t\_errorprogramp2(v\_errorCountp2) := i\_scheduleName;  IF (v\_errorCountp2 = C\_ERRORCOUNTP2) THEN  GOTO insertzdoep2;  END IF;  END IF;  END IF;  \*/  --MPS  ----EFFDATE Validation  ----------------MB2:policy not migrated----  IF ( TRIM(obj\_mbrindp2.chdrnum\_b) IS NULL ) THEN  v\_isanyerrorp2 := 'Y';  v\_errorcountp2 := v\_errorcountp2 + 1;  i\_zdoe\_infop2.i\_indic := 'E';  i\_zdoe\_infop2.i\_error01 := 'PLNM';  i\_zdoe\_infop2.i\_errormsg01 := 'Policy Not migrated';  --i\_zdoe\_infop2.i\_errorprogram01 := i\_scheduleName;  i\_zdoe\_infop2.i\_errorprogram01 := 'BQ9SC';  pkg\_dm\_common\_operations.insertintozdoe(i\_zdoe\_info => i\_zdoe\_infop2);  CONTINUE skiprecordp2;  END IF;  --------------MB2:policy not migrated-------------  v\_iseffdate := validate\_date(trim(obj\_mbrindp2.effdate));  IF ( ( TRIM(obj\_mbrindp2.effdate) IS NULL ) OR ( trim(v\_iseffdate) <> 'OK' ) ) THEN  v\_isanyerrorp2 := 'Y';  v\_globalerrorp2 := 'Y';  v\_errorcountp2 := v\_errorcountp2 + 1;  t\_ercodep2(v\_errorcountp2) := c\_z013;  t\_errorfieldp2(v\_errorcountp2) := 'EFFDATE';  t\_errormsgp2(v\_errorcountp2) := o\_errortext(c\_z013);  t\_errorfieldvalp2(v\_errorcountp2) := obj\_mbrindp2.effdate;  t\_errorprogramp2(v\_errorcountp2) := i\_schedulename;  IF ( v\_errorcountp2 = c\_errorcountp2 ) THEN  GOTO insertzdoep2;  END IF;  END IF;  ----HSUMINSU Validation  /\* IF (TRIM(obj\_mbrindp2.hsuminsu) = C\_ZERO) THEN  v\_isAnyErrorp2 := 'Y';  v\_GlobalErrorp2 := 'Y';  v\_errorCountp2 := v\_errorCountp2 + 1;  t\_ercodep2(v\_errorCountp2) := C\_Z092;  t\_errorfieldp2(v\_errorCountp2) := 'HSUMINSU';  t\_errormsgp2(v\_errorCountp2) := o\_errortext(C\_Z092);  t\_errorfieldvalp2(v\_errorCountp2) := obj\_mbrindp2.hsuminsu;  t\_errorprogramp2(v\_errorCountp2) := i\_scheduleName;  IF (v\_errorCountp2 = C\_ERRORCOUNTP2) THEN  GOTO insertzdoep2;  END IF;  END IF; \*/  ---------------second part of validation -TITDMGMBRINDP2----------------------------------------  --PRODTYP validation  IF TRIM(obj\_mbrindp2.prodtyp) IS NOT NULL THEN  IF NOT ( itemexist.EXISTS(trim('T9797')  || trim(obj\_mbrindp2.prodtyp)  || 1) ) THEN  v\_isanyerrorp2 := 'Y';  v\_globalerrorp2 := 'Y';  v\_errorcountp2 := v\_errorcountp2 + 1;  t\_ercodep2(v\_errorcountp2) := c\_z014;  t\_errorfieldp2(v\_errorcountp2) := 'PRODTYP';  t\_errormsgp2(v\_errorcountp2) := o\_errortext(c\_z014);  t\_errorfieldvalp2(v\_errorcountp2) := obj\_mbrindp2.prodtyp;  t\_errorprogramp2(v\_errorcountp2) := i\_schedulename;  IF ( v\_errorcountp2 = c\_errorcountp2 ) THEN  GOTO insertzdoep2;  END IF;  END IF;  END IF;  /\*\*\*\* ITR-4\_LOT2 : MB6 - MOD : condition change due to new requirement : START \*\*\*\*/  IF ( trim(obj\_mbrindp2.ztaxflg) <> 'Y' ) AND ( trim(obj\_mbrindp2.ztaxflg) <> 'N' ) THEN  v\_isanyerrorp2 := 'Y';  v\_globalerrorp2 := 'Y';  v\_errorcountp2 := v\_errorcountp2 + 1;  t\_ercodep2(v\_errorcountp2) := c\_z028;  t\_errorfieldp2(v\_errorcountp2) := 'ZTAXFLG';  t\_errormsgp2(v\_errorcountp2) := o\_errortext(c\_z028);  t\_errorfieldvalp2(v\_errorcountp2) := obj\_mbrindp2.ztaxflg;  t\_errorprogramp2(v\_errorcountp2) := i\_schedulename;  IF ( v\_errorcountp2 = c\_errorcountp2 ) THEN  GOTO insertzdoep2;  END IF;  END IF;  /\*\*\*\* ITR-4\_LOT2 : MB6 - MOD : condition change due to new requirement : END \*\*\*\*/  -------------------------------Insert into ZDOEPF for P2 : START---------------------------------------  << insertzdoep2 >> IF ( v\_isanyerrorp2 = 'Y' ) THEN  IF TRIM(t\_ercodep2(1)) IS NOT NULL THEN  i\_zdoe\_infop2.i\_indic := c\_error;  i\_zdoe\_infop2.i\_error01 := t\_ercodep2(1);  i\_zdoe\_infop2.i\_errormsg01 := t\_errormsgp2(1);  i\_zdoe\_infop2.i\_errorfield01 := t\_errorfieldp2(1);  i\_zdoe\_infop2.i\_fieldvalue01 := t\_errorfieldvalp2(1);  i\_zdoe\_infop2.i\_errorprogram01 := t\_errorprogramp2(1);  END IF;  IF TRIM(t\_ercodep2(2)) IS NOT NULL THEN  i\_zdoe\_infop2.i\_indic := c\_error;  i\_zdoe\_infop2.i\_error02 := t\_ercodep2(2);  i\_zdoe\_infop2.i\_errormsg02 := t\_errormsgp2(2);  i\_zdoe\_infop2.i\_errorfield02 := t\_errorfieldp2(2);  i\_zdoe\_infop2.i\_fieldvalue02 := t\_errorfieldvalp2(2);  i\_zdoe\_infop2.i\_errorprogram02 := t\_errorprogramp2(2);  END IF;  IF TRIM(t\_ercodep2(3)) IS NOT NULL THEN  i\_zdoe\_infop2.i\_indic := c\_error;  i\_zdoe\_infop2.i\_error03 := t\_ercodep2(3);  i\_zdoe\_infop2.i\_errormsg03 := t\_errormsgp2(3);  i\_zdoe\_infop2.i\_errorfield03 := t\_errorfieldp2(3);  i\_zdoe\_infop2.i\_fieldvalue03 := t\_errorfieldvalp2(3);  i\_zdoe\_infop2.i\_errorprogram03 := t\_errorprogramp2(3);  END IF;  IF TRIM(t\_ercodep2(4)) IS NOT NULL THEN  i\_zdoe\_infop2.i\_indic := c\_error;  i\_zdoe\_infop2.i\_error04 := t\_ercodep2(4);  i\_zdoe\_infop2.i\_errormsg04 := t\_errormsgp2(4);  i\_zdoe\_infop2.i\_errorfield04 := t\_errorfieldp2(4);  i\_zdoe\_infop2.i\_fieldvalue04 := t\_errorfieldvalp2(4);  i\_zdoe\_infop2.i\_errorprogram04 := t\_errorprogramp2(4);  END IF;  IF TRIM(t\_ercodep2(5)) IS NOT NULL THEN  i\_zdoe\_infop2.i\_indic := c\_error;  i\_zdoe\_infop2.i\_error05 := t\_ercodep2(5);  i\_zdoe\_infop2.i\_errormsg05 := t\_errormsgp2(5);  i\_zdoe\_infop2.i\_errorfield05 := t\_errorfieldp2(5);  i\_zdoe\_infop2.i\_fieldvalue05 := t\_errorfieldvalp2(5);  i\_zdoe\_infop2.i\_errorprogram05 := t\_errorprogramp2(5);  END IF;  pkg\_dm\_common\_operations.insertintozdoe(i\_zdoe\_info => i\_zdoe\_infop2);  CONTINUE skiprecordp2;  END IF;  IF ( v\_isanyerrorp2 = 'N' ) THEN  i\_zdoe\_infop2.i\_indic := 'S';  pkg\_dm\_common\_operations.insertintozdoe(i\_zdoe\_info => i\_zdoe\_infop2);  END IF;  -------------------------------Insert into ZDOEPF for P2 : END---------------------------------------  ------------------INSERT INTO IG tables and registry table---------------------------------------------  IF ( ( trim(i\_zprvaldyn) = 'N' ) AND ( trim(v\_isanyerrorp2) = 'N' ) ) THEN  -- i\_text := TRIM(obj\_mbrindp1.mpolnum);  -- dmlog\_info(i\_lkey => 'obj\_mbrindp1.mpolnum', i\_ltext => i\_text);  /\*IF (TRIM(obj\_mbrindp1.mpolnum) IS NOT NULL) THEN  select ZPLANCLS, Zcolmcls  into v\_zplancls, v\_zcolmcls  from VM1DTA.GCHPPF  where TRIM(CHDRNUM) = TRIM(obj\_mbrindp1.mpolnum)  and CHDRCOY = TRIM(i\_company);  v\_zplancls\_new := v\_zplancls;  ELSE  v\_zplancls\_new := 'PP';  END IF;  \*/  --- MB16 START --  IF ( trim(i\_ztrxstat) ) <> 'RJ' THEN  v\_trmdate := c\_maxdate;  IF ( trim(i\_zpoltdate) <> c\_maxdate ) AND ( TRIM(i\_zpoltdate) IS NOT NULL ) THEN  v\_trmdate := i\_zpoltdate;  END IF;  IF ( trim(obj\_mbrindp1.dtetrm) <> c\_maxdate ) AND ( TRIM(obj\_mbrindp1.dtetrm) IS NOT NULL ) THEN  v\_trmdate := i\_dtetrm;  END IF;  END IF;  IF ( TRIM(i\_mpolnum) IS NOT NULL ) THEN  obj\_getmpol := getmpol(trim(i\_mpolnum));  v\_zplancls := obj\_getmpol.zplancls;  v\_zcolmcls := obj\_getmpol.zcolmcls;  v\_zplancls\_new := v\_zplancls;  ELSE  v\_zplancls\_new := 'PP';  END IF;  ---Insert into IG table "GXHIPF " BEGIN (pq9ho.updateGxhipf())---  SELECT  seq\_gxhipf.NEXTVAL  INTO v\_seq\_gxhipf  FROM  dual;  obj\_gxhipf.unique\_number := v\_seq\_gxhipf;  obj\_gxhipf.chdrcoy := i\_company;  obj\_gxhipf.chdrnum := substr(trim(obj\_mbrindp2.refnum), 1, 8);  obj\_gxhipf.mbrno := o\_defaultvalues('MBRNO');  obj\_gxhipf.prodtyp := obj\_mbrindp2.prodtyp;  obj\_gxhipf.planno := o\_defaultvalues('PLANNO');  obj\_gxhipf.effdate := obj\_mbrindp2.effdate;  obj\_gxhipf.headno := c\_space;  obj\_gxhipf.fmlycde := c\_space;  obj\_gxhipf.dteatt := obj\_mbrindp2.effdate;  IF ( trim(i\_statcode) = 'CA' ) THEN  obj\_gxhipf.dtetrm := i\_dtetrm;  ELSE  obj\_gxhipf.dtetrm := c\_maxdate;  END IF;  -- obj\_gxhipf.DTETRM := C\_MAXDATE;  obj\_gxhipf.reasontrm := c\_space;  obj\_gxhipf.xcesssi := c\_zero;  obj\_gxhipf.aprvdate := c\_maxdate;  obj\_gxhipf.accptdte := c\_maxdate;  obj\_gxhipf.spectrm := c\_space;  obj\_gxhipf.extrprm := c\_zero;  obj\_gxhipf.suminsu := obj\_mbrindp2.hsuminsu;  obj\_gxhipf.usersi := c\_zero;  obj\_gxhipf.decflg := c\_space;  obj\_gxhipf.dpntno := o\_defaultvalues('DPNTNO');  -- obj\_gxhipf.TERMID := i\_vrcmTermid;  obj\_gxhipf.termid := c\_space; --MB7  obj\_gxhipf.trdt := i\_trdt;  obj\_gxhipf.trtm := i\_vrcmtime;  --IF (TRIM(i\_statcode) = 'CA') THEN  -- obj\_gxhipf.TRANNO := v\_seqnoincr;  -- ELSE  -- obj\_gxhipf.TRANNO := v\_seqno;  -- END IF;  obj\_gxhipf.tranno := 1;  obj\_gxhipf.emload := c\_zero;  obj\_gxhipf.oaload := c\_zero;  obj\_gxhipf.billactn := c\_space;  obj\_gxhipf.impaircd01 := c\_space;  obj\_gxhipf.impaircd02 := c\_space;  obj\_gxhipf.impaircd03 := c\_space;  obj\_gxhipf.riemload := c\_zero;  obj\_gxhipf.rioaload := c\_zero;  obj\_gxhipf.stdprmload := c\_zero;  obj\_gxhipf.dteclam := c\_zero;  obj\_gxhipf.usrprf := i\_usrprf;  obj\_gxhipf.jobnm := i\_schedulename;  obj\_gxhipf.datime := CAST ( SYSDATE AS TIMESTAMP );  IF ( trim(v\_zplancls\_new) = 'FP' ) THEN  obj\_gxhipf.dprem := obj\_mbrindp2.aprem;  END IF;  IF ( trim(v\_zplancls\_new) = 'PP' ) THEN  obj\_gxhipf.dprem := trim(obj\_mbrindp2.aprem) / 12;  END IF;  obj\_gxhipf.loadreason := c\_space;  obj\_gxhipf.mbrind := c\_space;  obj\_gxhipf.ncbsi := 0;  -- obj\_gxhipf.RIPROCDT := C\_MAXDATE;  --obj\_gxhipf.RIPROCDT := C\_ZERO; --MB7  IF obj\_gxhipf.dtetrm = c\_maxdate THEN  obj\_gxhipf.riprocdt := c\_zero;  ELSE  obj\_gxhipf.riprocdt := c\_maxdate;  END IF;  -- obj\_gxhipf.USER\_T := i\_vrcmuser;  obj\_gxhipf.user\_t := c\_zero; --MB7  IF ( i\_zplancde IS NOT NULL ) THEN  /\* select ZINSTYPE  into v\_zinstype  from ZSLPPF  where TRIM(zsalplan) = TRIM(zplancde)  and TRIM(prodtyp) = TRIM(obj\_mbrindp2.prodtyp);  obj\_gxhipf.ZINSTYPE := v\_zinstype;\*/  IF ( getzinstype.EXISTS(trim(i\_zplancde)  || trim(obj\_mbrindp2.prodtyp)) ) THEN  obj\_gxhipf.zinstype := getzinstype(trim(i\_zplancde)  || trim(obj\_mbrindp2.prodtyp)); -- obj\_gxhipf.ZINSTYP := 'a';  END IF;  IF ( ( TRIM(i\_zwaitpedt) IS NOT NULL ) OR ( trim(i\_zwaitpedt) != 0 ) OR ( trim(i\_zwaitpedt) <> c\_maxdate ) )  THEN  IF ( TRIM(obj\_mbrindp2.prodtyp) IN (  '1989',  '1990',  '1991',  '1992',  '1993',  '1995',  '1996',  '1997'  ) ) THEN  obj\_gxhipf.zwaitpedt := i\_zwaitpedt;  ELSE  obj\_gxhipf.zwaitpedt := c\_maxdate;  END IF;  ELSE  obj\_gxhipf.zwaitpedt := c\_maxdate;  END IF;  END IF;  /\*\*\*\* ITR-4\_LOT2 : MB6 - MOD : condition change due to new requirement : START \*\*\*\*/  obj\_gxhipf.ztaxflg := obj\_mbrindp2.ztaxflg;  /\*\*\*\* ITR-4\_LOT2 : Mb6 - MOD : condition change due to new requirement : END \*\*\*\*/  -- insert into GXHIPF values obj\_gxhipf;  obj\_gxhipf.aprvdate := obj\_gxhipf.dteatt;  obj\_gxhipf.accptdte := obj\_gxhipf.dtetrm;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO GXHIPF VALUES obj\_gxhipf;  ---MB18  gxhipfindex := gxhipfindex + 1;  gxhipf\_list.extend;  gxhipf\_list(gxhipfindex) := obj\_gxhipf;  ---MB18  --------insert into IG table "GMOVPF" for GXHIPF BEGIN(pq9ho.updateGmhdpf())----  -- IF (TRIM(i\_statcode) <> 'PN') THEN Previous code  IF ( trim(i\_statcode) = 'XN' ) THEN  v\_olddta3 := obj\_gxhipf.chdrcoy  || obj\_gxhipf.chdrnum  || obj\_gxhipf.prodtyp  || obj\_gxhipf.planno  || obj\_gxhipf.effdate  || obj\_gxhipf.mbrno  || obj\_gxhipf.dpntno;  v\_refkey := obj\_gxhipf.chdrcoy  || obj\_gxhipf.chdrnum  || obj\_gxhipf.prodtyp  || obj\_gxhipf.planno  || obj\_gxhipf.effdate  || obj\_gxhipf.mbrno  || obj\_gxhipf.dpntno;  obj\_gmovpf3pn.chdrcoy := obj\_gxhipf.chdrcoy;  obj\_gmovpf3pn.chdrnum := obj\_gxhipf.chdrnum;  obj\_gmovpf3pn.effdate := i\_effdate;  obj\_gmovpf3pn.tranno := 1;  obj\_gmovpf3pn.user\_t := obj\_gxhipf.user\_t;  obj\_gmovpf3pn.trdt := obj\_gxhipf.trdt;  obj\_gmovpf3pn.trtm := obj\_gxhipf.trtm;  obj\_gmovpf3pn.termid := obj\_gxhipf.termid;  obj\_gmovpf3pn.batccoy := i\_company;  obj\_gmovpf3pn.batcbrn := i\_branch;  obj\_gmovpf3pn.batcactyr := i\_acctyear;  obj\_gmovpf3pn.batcactmn := i\_acctmonth;  IF ( trim(i\_statcode) = 'CA' ) THEN  obj\_gmovpf3pn.batctrcde := 'T913';  ELSE  obj\_gmovpf3pn.batctrcde := 'T903';  END IF;  obj\_gmovpf3pn.batcbatch := c\_space;  obj\_gmovpf3pn.olddta := v\_olddta3;  obj\_gmovpf3pn.newdta := '';  obj\_gmovpf3pn.funccode := c\_funcdeaddmbrpp;  obj\_gmovpf3pn.refkey := v\_refkey;  obj\_gmovpf3pn.rfmt := 'GXHIPEMREC';  obj\_gmovpf3pn.chgtype := '';  obj\_gmovpf3pn.riprior := ' ';  obj\_gmovpf3pn.onpreflg := '';  obj\_gmovpf3pn.fachold := '';  obj\_gmovpf3pn.planvflg := '';  obj\_gmovpf3pn.usrprf := i\_usrprf;  obj\_gmovpf3pn.jobnm := i\_schedulename;  obj\_gmovpf3pn.datime := CAST ( SYSDATE AS TIMESTAMP );  SELECT  seq\_gmovpf.NEXTVAL  INTO v\_seq\_gmovpf  FROM  dual;  obj\_gmovpf3pn.unique\_number := v\_seq\_gmovpf;  -- Insert into GMOVPF values obj\_gmovpf3PN;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO gmovpf VALUES obj\_gmovpf3PN;  ---MB18  gmovpf3pnindex := gmovpf3pnindex + 1;  gmovpf3pn\_list.extend;  gmovpf3pn\_list(gmovpf3pnindex) := obj\_gmovpf3pn;  ---MB18  END IF;  /\* IF (TRIM(i\_statcode) = 'CA') THEN  v\_olddta3 := obj\_gxhipf.Chdrcoy || obj\_gxhipf.Chdrnum ||  obj\_gxhipf.Prodtyp || obj\_gxhipf.Planno ||  obj\_gxhipf.Effdate || obj\_gxhipf.Mbrno ||  obj\_gxhipf.Dpntno;  v\_newdta3 := obj\_gxhipf.Chdrcoy || obj\_gxhipf.Chdrnum ||  obj\_gxhipf.Prodtyp || obj\_gxhipf.Planno ||  obj\_gxhipf.Effdate || obj\_gxhipf.Mbrno ||  obj\_gxhipf.Dpntno;  v\_refkey := obj\_gxhipf.Chdrcoy || obj\_gxhipf.Chdrnum ||  obj\_gxhipf.Prodtyp || obj\_gxhipf.Planno ||  obj\_gxhipf.Effdate || obj\_gxhipf.Mbrno ||  obj\_gxhipf.Dpntno;  obj\_gmovpf3CA.Chdrcoy := obj\_gxhipf.Chdrcoy;  obj\_gmovpf3CA.Chdrnum := obj\_gxhipf.Chdrnum;  obj\_gmovpf3CA.Effdate := obj\_gxhipf.effdate;  obj\_gmovpf3CA.Tranno := 2;  obj\_gmovpf3CA.User\_t := obj\_gxhipf.User\_t;  obj\_gmovpf3CA.Trdt := obj\_gxhipf.Trdt;  obj\_gmovpf3CA.Trtm := obj\_gxhipf.Trtm;  obj\_gmovpf3CA.Termid := obj\_gxhipf.Termid;  obj\_gmovpf3CA.Batccoy := i\_company;  obj\_gmovpf3CA.Batcbrn := i\_branch;  obj\_gmovpf3CA.Batcactyr := i\_acctYear;  obj\_gmovpf3CA.Batcactmn := i\_acctMonth;  IF (TRIM(i\_statcode) = 'CA') THEN  obj\_gmovpf3CA.Batctrcde := 'T913';  ELSE  obj\_gmovpf3CA.Batctrcde := 'T903';  END IF;  obj\_gmovpf3CA.Batcbatch := C\_SPACE;  obj\_gmovpf3CA.Olddta := v\_olddta3;  obj\_gmovpf3CA.Newdta := v\_newdta3;  obj\_gmovpf3CA.Funccode := C\_FUNCDETRMBRPP;  obj\_gmovpf3CA.Refkey := v\_refkey;  obj\_gmovpf3CA.Rfmt := 'GXHIPEMREC';  obj\_gmovpf3CA.Chgtype := '';  obj\_gmovpf3CA.Riprior := ' ';  obj\_gmovpf3CA.Onpreflg := '';  obj\_gmovpf3CA.Fachold := '';  obj\_gmovpf3CA.Planvflg := '';  obj\_gmovpf3CA.Usrprf := i\_usrprf;  obj\_gmovpf3CA.Jobnm := i\_scheduleName;  obj\_gmovpf3CA.Datime := CAST(sysdate AS TIMESTAMP);  -- Insert into GMOVPF values obj\_gmovpf3CA;  gmovpf3CAindex := gmovpf3CAindex + 1;  gmovpf3CA\_list.extend;  gmovpf3CA\_list(gmovpf3CAindex) := obj\_gmovpf3CA;  END IF;\*/  --------insert into IG table "GMOVPF" for GXHIPF END (pq9ho.updateGmhdpf())----  ---Insert into IG table "GXHIPF" END ---  ---Insert into IG table "GAPHPF" BEGIN ---  SELECT  seq\_gaphpf.NEXTVAL  INTO v\_seq\_gaphpf  FROM  dual;  obj\_gaphpf.unique\_number := v\_seq\_gaphpf;  obj\_gaphpf.chdrcoy := i\_company;  obj\_gaphpf.chdrnum := substr(trim(obj\_mbrindp2.refnum), 1, 8);  obj\_gaphpf.headcntind := o\_defaultvalues('HEADCNTIND');  obj\_gaphpf.mbrno := o\_defaultvalues('MBRNO');  obj\_gaphpf.dpntno := o\_defaultvalues('DPNTNO');  obj\_gaphpf.prodtyp := obj\_mbrindp2.prodtyp;  obj\_gaphpf.planno := o\_defaultvalues('PLANNO');  obj\_gaphpf.effdate := obj\_mbrindp2.effdate;  --MB16 F (TRIM(i\_statcode) = 'CA') THEN  --MB16 obj\_gaphpf.DTETRM := i\_dtetrm;  --MB16 ELSE  --MB16 obj\_gaphpf.DTETRM := C\_MAXDATE;  --MB16 END IF;  obj\_gaphpf.dtetrm := c\_maxdate;  --MB16 IF (TRIM(i\_statcode) = 'CA') THEN  --MB16 obj\_gaphpf.TRANNO := v\_seqnoincr;  --MB16 ELSE  --MB16 obj\_gaphpf.TRANNO := v\_seqno;  --MB16 END IF;  obj\_gaphpf.tranno := v\_seqno; --MB16  obj\_gaphpf.subscoy := c\_space;  obj\_gaphpf.subsnum := c\_space;  obj\_gaphpf.aprem := obj\_mbrindp2.aprem;  obj\_gaphpf.aextprm := c\_zero;  obj\_gaphpf.aextprmr := c\_zero;  obj\_gaphpf.hsuminsu := obj\_mbrindp2.hsuminsu;  obj\_gaphpf.prmamtrt := obj\_mbrindp2.aprem;  obj\_gaphpf.indicx := o\_defaultvalues('INDICX');  obj\_gaphpf.updtype := o\_defaultvalues('UPDTYPEN'); -- MB16  obj\_gaphpf.resncd := o\_defaultvalues('RESNCDA'); -- MB16  -- MB16 START --  --IF (TRIM(i\_statcode) = 'CA') THEN  -- obj\_gaphpf.UPDTYPE := o\_defaultvalues('UPDTYPET');  --ELSE  -- obj\_gaphpf.UPDTYPE := o\_defaultvalues('UPDTYPEN');  --END IF;  --IF (TRIM(i\_statcode) = 'CA') THEN  -- obj\_gaphpf.RESNCD := o\_defaultvalues('RESNCDT');  --ELSE  -- obj\_gaphpf.RESNCD := o\_defaultvalues('RESNCDA');  --END IF;  -- MB16 END --  obj\_gaphpf.srcdata := o\_defaultvalues('SRCDATA');  obj\_gaphpf.batctrcd := c\_newbz\_issue; -- MB16  --MB16 STARTS --  --IF ((TRIM(i\_statcode) = 'IF') OR (TRIM(i\_statcode) = 'XN')) THEN  -- obj\_gaphpf.BATCTRCD := C\_NEWBZ\_ISSUE;  --END IF;  --IF (TRIM(i\_statcode) = 'CA') THEN  -- obj\_gaphpf.BATCTRCD := C\_ISS\_TERM;  --END IF;  -- MB16 END --  obj\_gaphpf.validflag := o\_defaultvalues('VALIDFLAG');  obj\_gaphpf.jobnm := i\_schedulename;  obj\_gaphpf.usrprf := i\_usrprf;  obj\_gaphpf.datime := CAST ( SYSDATE AS TIMESTAMP );  -- Insert into GAPHPF values obj\_gaphpf;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO gaphpf VALUES obj\_gaphpf;  ---MB18  gaphpfindex := gaphpfindex + 1;  gaphpf\_list.extend;  gaphpf\_list(gaphpfindex) := obj\_gaphpf;  ---MB18  ---Insert into IG table "GAPHPF" END ---  ---Insert into IG table "MTRNPF" BEGIN ---  /\* IF (TRIM(i\_statcode) <> 'PN') THEN  obj\_mtrnpfPN.CHDRCOY := i\_company;  obj\_mtrnpfPN.CHDRNUM := SUBSTR(TRIM(obj\_mbrindp2.refnum), 1, 8);  obj\_mtrnpfPN.PRODTYP := obj\_mbrindp2.prodtyp;  obj\_mtrnpfPN.MBRNO := o\_defaultvalues('MBRNO');  obj\_mtrnpfPN.DPNTNO := o\_defaultvalues('DPNTNO');  obj\_mtrnpfPN.TRANNO := 1;  obj\_mtrnpfPN.RLDPNTNO := C\_SPACE;  obj\_mtrnpfPN.ISSDATE := obj\_mbrindp2.effdate;  obj\_mtrnpfPN.CHGTYPE := C\_ADDMBR;  obj\_mtrnpfPN.EFFDATE := obj\_mbrindp2.effdate;  obj\_mtrnpfPN.JOBNOTPA := c\_ZERO;  obj\_mtrnpfPN.JOBNORIE := c\_ZERO;  obj\_mtrnpfPN.USRPRF := i\_usrprf; -- obj\_mtrnpf.USER\_PROFILE := 'a';  obj\_mtrnpfPN.JOBNM := i\_scheduleName; -- obj\_mtrnpf.JOB\_NAME := 'a';  obj\_mtrnpfPN.DATIME := CAST(sysdate AS TIMESTAMP);  -- insert into MTRNPF values obj\_mtrnpfPN;  mtrnpfPNindex := mtrnpfPNindex + 1;  mtrnpfPN\_list.extend;  mtrnpfPN\_list(mtrnpfPNindex) := obj\_mtrnpfPN;  END IF;  IF (TRIM(i\_statcode) = 'CA') THEN  obj\_mtrnpfCA.CHDRCOY := i\_company;  obj\_mtrnpfCA.CHDRNUM := SUBSTR(TRIM(obj\_mbrindp2.refnum), 1, 8);  obj\_mtrnpfCA.PRODTYP := obj\_mbrindp2.prodtyp;  obj\_mtrnpfCA.MBRNO := o\_defaultvalues('MBRNO');  obj\_mtrnpfCA.DPNTNO := o\_defaultvalues('DPNTNO');  obj\_mtrnpfCA.TRANNO := 2;  obj\_mtrnpfCA.RLDPNTNO := C\_SPACE;  obj\_mtrnpfCA.ISSDATE := obj\_mbrindp2.effdate;  obj\_mtrnpfCA.CHGTYPE := C\_TRMPOL;  obj\_mtrnpfCA.EFFDATE := obj\_mbrindp2.effdate;  obj\_mtrnpfCA.JOBNOTPA := c\_ZERO;  obj\_mtrnpfCA.JOBNORIE := c\_ZERO;  -- obj\_mtrnpf.USER\_PROFILE := 'a';  -- obj\_mtrnpf.JOB\_NAME := 'a';  obj\_mtrnpfCA.DATIME := CAST(sysdate AS TIMESTAMP);  -- insert into MTRNPF values obj\_mtrnpfCA;  mtrnpfCAindex := mtrnpfCAindex + 1;  mtrnpfCA\_list.extend;  mtrnpfCA\_list(mtrnpfCAindex) := obj\_mtrnpfCA;  END IF;\*/  ---Insert into IG table "MTRNPF" END ---  ---Insert into IG table "ZTIERPF " START ---  obj\_ztierpf.chdrcoy := i\_company;  -- dbms\_output.put\_line('obj\_mbrindp2.refnum ==>' || obj\_mbrindp2.refnum);  obj\_ztierpf.chdrnum := substr(trim(obj\_mbrindp2.refnum), 1, 8);  obj\_ztierpf.mbrno := o\_defaultvalues('MBRNO');  obj\_ztierpf.dpntno := o\_defaultvalues('DPNTNO');  obj\_ztierpf.prodtyp := obj\_mbrindp2.prodtyp;  --IF (TRIM(i\_statcode) = 'CA') THEN -- MB16  --IF v\_trmdate <> C\_MAXDATE THEN --MB16  -- obj\_ztierpf.TRANNO := v\_seqnoincr;  --ELSE  -- obj\_ztierpf.TRANNO := v\_seqno;  --END IF;  -- obj\_ztierpf.TRANNO := v\_seqno;  -- obj\_ztierpf.EFFDATE := obj\_mbrindp2.effdate;--MB7 comment out as setting null  obj\_ztierpf.tranno := 1;  obj\_ztierpf.effdate := obj\_mbrindp2.effdate; --MB10  obj\_ztierpf.ztierno := o\_defaultvalues('ZTIERNO');  --IF (TRIM(i\_statcode) = 'CA') THEN -- MB16  IF v\_trmdate <> c\_maxdate THEN  --MB16  obj\_ztierpf.dtetrm := i\_dtetrm;  ELSE  obj\_ztierpf.dtetrm := c\_maxdate;  END IF;  obj\_ztierpf.dteatt := obj\_mbrindp2.effdate;  obj\_ztierpf.suminsu := obj\_mbrindp2.hsuminsu;  -- obj\_ztierpf.DPREM := obj\_mbrindp2.APREM;  IF ( trim(v\_zplancls\_new) = 'FP' ) THEN  obj\_ztierpf.dprem := obj\_mbrindp2.aprem;  END IF;  IF ( trim(v\_zplancls\_new) = 'PP' ) THEN  obj\_ztierpf.dprem := trim(obj\_mbrindp2.aprem) / 12;  END IF;  obj\_ztierpf.usrprf := i\_usrprf;  obj\_ztierpf.jobnm := i\_schedulename;  obj\_ztierpf.datime := CAST ( SYSDATE AS TIMESTAMP );  ----- MB5 START -----  --obj\_ztierpf.EFDATE := C\_MAXDATE; --- MB5  --obj\_ztierpf.ZMDDVDT := C\_MAXDATE; --- MB5  IF obj\_ztierpf.dtetrm <> c\_maxdate THEN  obj\_ztierpf.zreindt := c\_maxdate;  ELSE  obj\_ztierpf.zreindt := c\_zero;  END IF;  --Obj\_ztierpf.ZREINDT := C\_ZERO; --- MB5  obj\_ztierpf.zcvgstrtdt := obj\_ztierpf.dteatt;  obj\_ztierpf.zcvgenddt := obj\_ztierpf.dtetrm;  obj\_ztierpf.zwaitperd := obj\_gxhipf.zwaitpedt;  ----- MB5 END -----  --INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO VIEW\_DM\_ZTIERPF VALUES obj\_ztierpf;  ---MB18  ztierpfindex := ztierpfindex + 1;  ztierpf\_list.extend;  ztierpf\_list(ztierpfindex) := obj\_ztierpf;  ---MB18  ---Insert into IG table "ZTIERPF " END ---\*/  ---Insert into IG table "ZTEMPCOVPF" START ---  obj\_ztempcovpf1.chdrcoy := i\_company;  obj\_ztempcovpf1.chdrnum := substr(trim(obj\_mbrindp2.refnum), 1, 8);  obj\_ztempcovpf1.altquoteno := NULL;  obj\_ztempcovpf1.tranno := v\_seqno;  obj\_ztempcovpf1.mbrno := o\_defaultvalues('MBRNO');  obj\_ztempcovpf1.dpntno := o\_defaultvalues('DPNTNO');  obj\_ztempcovpf1.prodtyp := obj\_mbrindp2.prodtyp;  obj\_ztempcovpf1.effdate := obj\_mbrindp2.effdate;  obj\_ztempcovpf1.dteatt := obj\_mbrindp2.effdate;  /\*IF (TRIM(i\_statcode) = 'CA') THEN  obj\_ztempcovpf1.DTETRM := i\_dtetrm;  ELSE  obj\_ztempcovpf1.DTETRM := C\_MAXDATE;  END IF;\*/  obj\_ztempcovpf1.dtetrm := c\_maxdate;  obj\_ztempcovpf1.sumins := obj\_mbrindp2.hsuminsu;  -- obj\_ztempcovpf1.DPREM := obj\_mbrindp2.APREM;  IF ( trim(v\_zplancls\_new) = 'FP' ) THEN  obj\_ztempcovpf1.dprem := obj\_mbrindp2.aprem;  END IF;  IF ( trim(v\_zplancls\_new) = 'PP' ) THEN  obj\_ztempcovpf1.dprem := trim(obj\_mbrindp2.aprem) / 12;  END IF;  -- obj\_ztempcovpf1.ZWPENDDT := C\_MAXDATE;  IF ( ( TRIM(i\_zwaitpedt) IS NOT NULL ) OR ( trim(i\_zwaitpedt) != 0 ) OR ( trim(i\_zwaitpedt) <> c\_maxdate ) ) THEN  IF ( TRIM(obj\_mbrindp2.prodtyp) IN (  '1989',  '1990',  '1991',  '1992',  '1993',  '1995',  '1996',  '1997'  ) ) THEN  obj\_ztempcovpf1.zwpenddt := i\_zwaitpedt;  ELSE  obj\_ztempcovpf1.zwpenddt := c\_maxdate;  END IF;  ELSE  obj\_ztempcovpf1.zwpenddt := c\_maxdate;  END IF;  obj\_ztempcovpf1.zchgtype := 'A';  obj\_ztempcovpf1.dsumin := obj\_mbrindp2.hsuminsu;  obj\_ztempcovpf1.zsalplan := i\_zplancde;  obj\_ztempcovpf1.usrprf := i\_usrprf;  obj\_ztempcovpf1.jobnm := i\_schedulename;  obj\_ztempcovpf1.datime := CAST ( SYSDATE AS TIMESTAMP );  obj\_ztempcovpf1.zinstype := o\_defaultvalues('ZINSTYPE');  --obj\_ztempcovpf1.REFNUM := 'a';  obj\_ztempcovpf1.zcvgstrtdt := obj\_ztempcovpf1.dteatt;  obj\_ztempcovpf1.zcvgenddt := obj\_ztempcovpf1.dtetrm;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO VIEW\_DM\_ZTEMPCOVPF VALUES obj\_ztempcovpf1;  ---MB18  ztempcovpf1index := ztempcovpf1index + 1;  ztempcovpf1\_list.extend;  ztempcovpf1\_list(ztempcovpf1index) := obj\_ztempcovpf1;  ---MB18  --IF (TRIM(i\_statcode) = 'CA') THEN -- MB16  IF v\_trmdate <> c\_maxdate THEN  --MB16  obj\_ztempcovpf2.chdrcoy := i\_company;  obj\_ztempcovpf2.chdrnum := substr(trim(obj\_mbrindp2.refnum), 1, 8);  obj\_ztempcovpf2.altquoteno := NULL;  obj\_ztempcovpf2.tranno := v\_seqnoincr;  obj\_ztempcovpf2.mbrno := o\_defaultvalues('MBRNO');  obj\_ztempcovpf2.dpntno := o\_defaultvalues('DPNTNO');  obj\_ztempcovpf2.prodtyp := obj\_mbrindp2.prodtyp;  obj\_ztempcovpf2.effdate := obj\_mbrindp2.effdate;  obj\_ztempcovpf2.dteatt := obj\_mbrindp2.effdate;  /\* IF (TRIM(i\_statcode) = 'CA') THEN  obj\_ztempcovpf2.DTETRM := i\_dtetrm;  ELSE  obj\_ztempcovpf2.DTETRM := C\_MAXDATE;  END IF;\*/  obj\_ztempcovpf2.dtetrm := i\_dtetrm;  obj\_ztempcovpf2.sumins := obj\_mbrindp2.hsuminsu;  -- obj\_ztempcovpf2.DPREM := obj\_mbrindp2.APREM;  IF ( trim(v\_zplancls\_new) = 'FP' ) THEN  obj\_ztempcovpf2.dprem := obj\_mbrindp2.aprem;  END IF;  IF ( trim(v\_zplancls\_new) = 'PP' ) THEN  obj\_ztempcovpf2.dprem := trim(obj\_mbrindp2.aprem) / 12;  END IF;  --obj\_ztempcovpf2.ZWPENDDT := C\_MAXDATE;  IF ( ( TRIM(i\_zwaitpedt) IS NOT NULL ) OR ( trim(i\_zwaitpedt) != 0 ) OR ( trim(i\_zwaitpedt) <> c\_maxdate ) )  THEN  IF ( TRIM(obj\_mbrindp2.prodtyp) IN (  '1989',  '1990',  '1991',  '1992',  '1993',  '1995',  '1996',  '1997'  ) ) THEN  obj\_ztempcovpf2.zwpenddt := i\_zwaitpedt;  ELSE  obj\_ztempcovpf2.zwpenddt := c\_maxdate;  END IF;  ELSE  obj\_ztempcovpf2.zwpenddt := c\_maxdate;  END IF;  obj\_ztempcovpf2.zchgtype := 'T';  obj\_ztempcovpf2.dsumin := obj\_mbrindp2.hsuminsu;  obj\_ztempcovpf2.zsalplan := i\_zplancde;  obj\_ztempcovpf2.usrprf := i\_usrprf;  obj\_ztempcovpf2.jobnm := i\_schedulename;  obj\_ztempcovpf2.datime := CAST ( SYSDATE AS TIMESTAMP );  obj\_ztempcovpf2.zinstype := o\_defaultvalues('ZINSTYPE');  -- obj\_ztempcovpf2.REFNUM := 'a';  obj\_ztempcovpf2.zcvgstrtdt := obj\_ztempcovpf2.dteatt;  obj\_ztempcovpf2.zcvgenddt := obj\_ztempcovpf2.dtetrm;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO VIEW\_DM\_ZTEMPCOVPF VALUES obj\_ztempcovpf2;  ---MB18  ztempcovpf2index := ztempcovpf2index + 1;  ztempcovpf2\_list.extend;  ztempcovpf2\_list(ztempcovpf2index) := obj\_ztempcovpf2;  ---MB18  END IF;  ---Insert into IG table "ZTEMPCOVPF" END ---\*/  ---Insert into IG table "ztemptierpf" START ---  obj\_ztemptierpf1.chdrcoy := i\_company;  obj\_ztemptierpf1.chdrnum := substr(trim(obj\_mbrindp2.refnum), 1, 8);  obj\_ztemptierpf1.altquoteno := NULL;  obj\_ztemptierpf1.mbrno := o\_defaultvalues('MBRNO');  obj\_ztemptierpf1.dpntno := o\_defaultvalues('DPNTNO');  obj\_ztemptierpf1.prodtyp := obj\_mbrindp2.prodtyp;  obj\_ztemptierpf1.tranno := v\_seqno;  obj\_ztemptierpf1.effdate := obj\_mbrindp2.effdate;  obj\_ztemptierpf1.ztierno := 1;  /\* IF (TRIM(i\_statcode) = 'CA') THEN  obj\_ztemptierpf1.DTETRM := i\_dtetrm;  ELSE  obj\_ztemptierpf1.DTETRM := C\_MAXDATE;  END IF;\*/  obj\_ztemptierpf1.dtetrm := c\_maxdate;  -- obj\_ztemptierpf1.DTETRM := 'a';  obj\_ztemptierpf1.dteatt := obj\_mbrindp2.effdate;  obj\_ztemptierpf1.suminsu := obj\_mbrindp2.hsuminsu;  -- obj\_ztemptierpf1.DPREM := obj\_mbrindp2.APREM;  IF ( trim(v\_zplancls\_new) = 'FP' ) THEN  obj\_ztemptierpf1.dprem := obj\_mbrindp2.aprem;  END IF;  IF ( trim(v\_zplancls\_new) = 'PP' ) THEN  obj\_ztemptierpf1.dprem := trim(obj\_mbrindp2.aprem) / 12;  END IF;  obj\_ztemptierpf1.zvioltyp := NULL;  --obj\_ztemptierpf1.EFDATE := NULL;--MB7 comment out as setting null  -- obj\_ztemptierpf1.ZMDDVDT := NULL;--MB7 comment out as setting null  obj\_ztemptierpf1.usrprf := i\_usrprf;  obj\_ztemptierpf1.jobnm := i\_schedulename;  obj\_ztemptierpf1.datime := CAST ( SYSDATE AS TIMESTAMP );  --obj\_ztemptierpf1.REFNUM := 'a';  obj\_ztemptierpf1.xtranno := 1;  obj\_ztemptierpf1.zchgtype := 'A';  ----- MB5 START -----  --obj\_ztemptierpf1.EFDATE := C\_MAXDATE;  --obj\_ztemptierpf1.ZMDDVDT := C\_MAXDATE;  ----- MB5 END -----  -- obj\_ztemptierpf1.ZREINDT := C\_MAXDATE;  obj\_ztemptierpf1.zcvgstrtdt := obj\_ztemptierpf1.dteatt;  obj\_ztemptierpf1.zcvgenddt := obj\_ztemptierpf1.dtetrm;  obj\_ztemptierpf1.zwaitperd := obj\_gxhipf.zwaitpedt;  --INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO VIEW\_DM\_ZTEMPTIERPF VALUES obj\_ztemptierpf1;  ---MB18  ztemptierpf1index := ztemptierpf1index + 1;  ztemptierpf1\_list.extend;  ztemptierpf1\_list(ztemptierpf1index) := obj\_ztemptierpf1;  ---MB18  --IF (TRIM(i\_statcode) = 'CA') THEN -- MB16  IF v\_trmdate <> c\_maxdate THEN  --MB16  obj\_ztemptierpf2.chdrcoy := i\_company;  obj\_ztemptierpf2.chdrnum := substr(trim(obj\_mbrindp2.refnum), 1, 8);  obj\_ztemptierpf2.altquoteno := NULL;  obj\_ztemptierpf2.mbrno := o\_defaultvalues('MBRNO');  obj\_ztemptierpf2.dpntno := o\_defaultvalues('DPNTNO');  obj\_ztemptierpf2.prodtyp := obj\_mbrindp2.prodtyp;  obj\_ztemptierpf2.tranno := v\_seqnoincr;  obj\_ztemptierpf2.effdate := obj\_mbrindp2.effdate; -- MB16  -- obj\_ztemptierpf2.EFFDATE := obj\_mbrindp2.EFFDATE; --MB7 comment out as setting null  obj\_ztemptierpf2.ztierno := 1;  /\*IF (TRIM(i\_statcode) = 'CA') THEN  obj\_ztemptierpf2.DTETRM := i\_dtetrm;  ELSE  obj\_ztemptierpf2.DTETRM := C\_MAXDATE;  END IF;\*/  obj\_ztemptierpf2.dtetrm := i\_dtetrm;  obj\_ztemptierpf2.dteatt := obj\_mbrindp2.effdate;  obj\_ztemptierpf2.suminsu := obj\_mbrindp2.hsuminsu;  -- obj\_ztemptierpf2.DPREM := obj\_mbrindp2.APREM;  IF ( trim(v\_zplancls\_new) = 'FP' ) THEN  obj\_ztemptierpf2.dprem := obj\_mbrindp2.aprem;  END IF;  IF ( trim(v\_zplancls\_new) = 'PP' ) THEN  obj\_ztemptierpf2.dprem := trim(obj\_mbrindp2.aprem) / 12;  END IF;  obj\_ztemptierpf2.zvioltyp := NULL;  -- obj\_ztemptierpf2.EFDATE := NULL;  -- obj\_ztemptierpf2.ZMDDVDT := NULL;--MB7 comment out as setting null  obj\_ztemptierpf2.usrprf := i\_usrprf;  obj\_ztemptierpf2.jobnm := i\_schedulename;  obj\_ztemptierpf2.datime := CAST ( SYSDATE AS TIMESTAMP );  -- obj\_ztemptierpf2.REFNUM := 'a';  obj\_ztemptierpf2.xtranno := 1;  obj\_ztemptierpf2.zchgtype := 'T';  ----- MB5 START -----  --obj\_ztemptierpf2.EFDATE := C\_MAXDATE;  --obj\_ztemptierpf2.ZMDDVDT := C\_MAXDATE;  ----- MB5 END -----  -- obj\_ztemptierpf2.ZREINDT := C\_MAXDATE;  obj\_ztemptierpf2.zcvgstrtdt := obj\_ztemptierpf2.dteatt;  obj\_ztemptierpf2.zcvgenddt := obj\_ztemptierpf2.dtetrm;  obj\_ztemptierpf2.zwaitperd := obj\_gxhipf.zwaitpedt;  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO VIEW\_DM\_ZTEMPTIERPF VALUES obj\_ztemptierpf2;  ---MB18  ztemptierpf2index := ztemptierpf2index + 1;  ztemptierpf2\_list.extend;  ztemptierpf2\_list(ztemptierpf2index) := obj\_ztemptierpf2;  ---MB18  END IF;  ---Insert into IG table "ztemptierpf" END ---\*/  END IF;  END LOOP;  idx := gxhipf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gxhipf\_list.first..gxhipf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gxhipf VALUES gxhipf\_list ( idx );  END IF;  idx := gmovpf3pn\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gmovpf3pn\_list.first..gmovpf3pn\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gmovpf VALUES gmovpf3pn\_list ( idx );  END IF;  idx := gaphpf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN gaphpf\_list.first..gaphpf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.gaphpf VALUES gaphpf\_list ( idx );  END IF;  idx := ztierpf\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN ztierpf\_list.first..ztierpf\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.view\_dm\_ztierpf VALUES ztierpf\_list ( idx );  END IF;  idx := ztempcovpf1\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN ztempcovpf1\_list.first..ztempcovpf1\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.view\_dm\_ztempcovpf VALUES ztempcovpf1\_list ( idx );  END IF;  idx := ztempcovpf2\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN ztempcovpf2\_list.first..ztempcovpf2\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.view\_dm\_ztempcovpf VALUES ztempcovpf2\_list ( idx );  END IF;  idx := ztemptierpf1\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN ztemptierpf1\_list.first..ztemptierpf1\_list.last  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.view\_dm\_ztemptierpf VALUES ztemptierpf1\_list ( idx );  END IF;  idx := ztemptierpf2\_list.first;  IF ( idx IS NOT NULL ) THEN  FORALL idx IN ztemptierpf2\_list.first..ztemptierpf2\_list.last  -- INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO VM1dta.View\_Dm\_Ztemptierpf VALUES ztemptierpf2\_list (idx);  INSERT /\*+ ENABLE\_PARALLEL\_DML \*/ INTO vm1dta.view\_dm\_ztemptierpf VALUES ztemptierpf2\_list ( idx );  END IF;  EXIT WHEN cur\_mbr\_ind\_p2%notfound;  gxhipf\_list.DELETE;  gmovpf3pn\_list.DELETE;  gaphpf\_list.DELETE;  ztierpf\_list.DELETE;  ztempcovpf1\_list.DELETE;  ztempcovpf2\_list.DELETE;  ztemptierpf1\_list.DELETE;  ztemptierpf2\_list.DELETE;  gxhipfindex := 0;  gmovpf3pnindex := 0;  gaphpfindex := 0;  ztierpfindex := 0;  ztempcovpf1index := 0;  ztempcovpf2index := 0;  ztemptierpf1index := 0;  ztemptierpf2index := 0;  END LOOP;  CLOSE cur\_mbr\_ind\_p2;  dbms\_output.put\_line('Procedure execution time = '  ||(dbms\_utility.get\_time - v\_timestart) / 100);  INSERT INTO vm1dta.dmbmonpf (  batch\_name,  start\_time,  end\_time,  job\_num,  datime,  remarks  ) VALUES (  i\_schedulename,  v\_timestart,  dbms\_utility.get\_time,  i\_schedulenumber,  SYSDATE,  start\_id  || '-'  || end\_id  );  --DBMS\_PROFILER.stop\_profiler;  END; |

Please refer below useful queries :

|  |
| --- |
| ALTER TRIGGER vm1dta.TR\_AUDIT\_CLEXPF DISABLE;  ALTER TRIGGER vm1dta.TR\_CLEXPF DISABLE;  ALTER TRIGGER vm1dta.TR\_CLRRPF DISABLE;  ALTER TRIGGER vm1dta.TR\_AUDIT\_CLRRPF DISABLE;  ALTER TRIGGER vm1dta.TR\_AUDIT\_CLNT DISABLE;  ALTER TRIGGER vm1dta.TR\_AUDIT\_CLNTPF DISABLE;  ALTER SEQUENCE VM1DTA.SEQ\_CLEXPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_CLRRPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_CLNTPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_ZDOEPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQANUMPF CACHE 20;  ----------------------------------------------------------  delete from vm1dta.ZDCLPF where prefix = 'CP';  create table vm1dta.ZDCLPF\_1412 as (select \* from vm1dta.ZDCLPF);  select \* from VM1DTA.ZDOECP0018 where TRIM(EROR01) is null;    select count(1) from vm1dta.clntpf where jobnm = 'NORMALPRC1';  select count(1) from vm1dta.clexpf where jobnm = 'NORMALPROC';  select count(1) from vm1dta.clrrpf where jobnm = 'NORMALPROC';  select count(1) from vm1dta.audit\_clrrpf where newjobnm = 'NORMALPROC';  delete from vm1dta.clntpf where jobnm = 'NORMALPROC';  delete from vm1dta.clexpf where jobnm = 'NORMALPROC';  delete from vm1dta.clrrpf where jobnm = 'NORMALPROC';  delete from vm1dta.audit\_clrrpf where newjobnm = 'NORMALPROC';  SELECT vm1dta.SEQANUMPF.nextval FROM dual;  select count(1) from vm1dta.clntpf where jobnm = 'FORALLPRC1';  select count(1) from vm1dta.clexpf where jobnm = 'FORALLPROC';  select count(1) from vm1dta.clrrpf where jobnm = 'FORALLPROC';  select count(1) from vm1dta.audit\_clrrpf where newjobnm = 'FORALLPROC';    -------------------------------------------------------------------------  select count(\*) from STAGEDBUSR.TITDMGMBRINDP1 ;  select count(\*) from STAGEDBUSR.TITDMGMBRINDP2;  SELECT \*  FROM TITDMGMBRINDP1@DMSTAGEDBLINK  where substr(refnum, 1, 8) ='00680125';    select \* from (SELECT A.\*, B.CHDRNUM AS CHDRNUM\_B  FROM TITDMGMBRINDP2@DMSTAGEDBLINK A  LEFT OUTER JOIN VM1DTA.ZDRPPF B  ON SUBSTR(A.refnum, 1, 8) = TRIM(B.CHDRNUM))  where SUBSTR(refnum, 1, 8) ='00680125';    select \* from ZDOEMB00 where indic ='S';  select \* from zdoein0009;  ---p1  select \* from STAGEDBUSR.TITDMGMBRINDP1 ;  select \* from vm1dta.GCHD r  where TRIM(chdrnum) IN ('00680125','00680435');  select \*  from vm1dta.GCHPPF r  where TRIM(chdrnum) IN ('00680125','00680435');  select \*  from vm1dta.GCHIPF r  where TRIM(chdrnum)IN ('00680125','00680435');  select \*  from vm1dta.GMHDPF r  where TRIM(chdrnum) IN ('00680125','00680435');  select \*  from vm1dta.GMHIPF r  where TRIM(chdrnum) IN ('00680125','00680435');  select \* from vm1dta.gpsupf where TRIM(chdrnum) IN ('00680125','00680435');  select \* from vm1dta.zcelinkpf where clntnum IN (select cownnum from vm1dta.gchd where chdrnum IN ('00680125','00680435'));  select \*  from vm1dta.zclepf where clntnum IN (select cownnum from vm1dta.gchd where chdrnum IN ('00680125','00680435'));  ----P2  select \*  from vm1dta.GXHIPF r  where TRIM(chdrnum) IN ('00680125','00680435');  select \*  from vm1dta.GMOVPF r  where TRIM(chdrnum) IN ('00680125','00680435');  select \*  from vm1dta.GAPHPF r  where TRIM(chdrnum) IN ('00680125','00680435');  select \*  from vm1dta.ztierpf r  where TRIM(chdrnum) IN ('00680125','00680435');  select \*  from vm1dta.ztempcovpf r  where TRIM(chdrnum) IN ('00680125','00680435');  select \*  from vm1dta.ztemptierpf r  where TRIM(chdrnum) IN ('00680125','00680435');  select \* from vm1dta.ZDRPPF where TRIM(chdrnum) IN ('00680125','00680435');  ---Delete a mig policy--  Delete from vm1dta.GCHD r  where TRIM(chdrnum) IN ('00680125','00680435');  Delete  from vm1dta.GCHPPF r  where TRIM(chdrnum)IN ('00680125','00680435');  Delete  from vm1dta.GCHIPF r  where TRIM(chdrnum) IN ('00680125','00680435');  Delete  from vm1dta.GMHDPF r  where TRIM(chdrnum) IN ('00680125','00680435');  Delete  from vm1dta.GMHIPF r  where TRIM(chdrnum) IN ('00680125','00680435');  Delete from vm1dta.gpsupf where TRIM(chdrnum) IN ('00680125','00680435');  Delete from vm1dta.zcelinkpf where clntnum IN (select cownnum from vm1dta.gchd where chdrnum IN ('00680125','00680435'));  Delete  from vm1dta.zclepf where clntnum IN (select cownnum from vm1dta.gchd where chdrnum IN ('00680125','00680435'));  ----P2  Delete  from vm1dta.GXHIPF r  where TRIM(chdrnum) IN ('00680125','00680435');  Delete  from vm1dta.GMOVPF r  where TRIM(chdrnum) IN ('00680125','00680435');  Delete  from vm1dta.GAPHPF r  where TRIM(chdrnum) IN ('00680125','00680435');  delete from  vm1dta.ztierpf r  where TRIM(chdrnum) IN ('00680125','00680435');  Delete  from vm1dta.ztempcovpf r  where TRIM(chdrnum) IN ('00680125','00680435');  Delete  from vm1dta.ztemptierpf r  where TRIM(chdrnum) IN ('00680125','00680435');  delete from vm1dta.ZDRPPF  where TRIM(chdrnum) IN ('00680125','00680435');  delete from vm1dta.ZDRPPF;  ALTER TRIGGER vm1dta.TR\_GCHPPF DISABLE;  ALTER TRIGGER vm1dta.TR\_GMHDPF DISABLE;  ALTER TRIGGER vm1dta.TR\_GMHIPF DISABLE;  ALTER TRIGGER vm1dta.TR\_GXHIPF DISABLE;  ALTER TRIGGER vm1dta.TR\_GAPHPF DISABLE;  ALTER TRIGGER vm1dta.TR\_GMOVPF DISABLE;  ALTER TRIGGER vm1dta.TR\_ZCLEPF DISABLE;  ALTER TRIGGER vm1dta.TR\_CHDRPF DISABLE;  ALTER TRIGGER vm1dta.TR\_GCHIPF DISABLE;  ALTER TRIGGER vm1dta.TR\_GPSUPF DISABLE;  select SEQ\_GPSUPF.CURRVAL from dual;  TR\_GPSUPF  select trigger\_name,Table\_name,status from All\_Triggers where table\_name IN (  'GCHD',  'CHDRPF',  'GCHPPF',  'GCHIPF',  'GMHDPF',  'GMHIPF',  'GPSUPF',  'ZCELINKPF',  'ZCLEPF',  'GXHIPF',  'GMOVPF',  'GAPHPF',  'ZTIERPF',  'ZTEMPCOVPF',  'ZTEMPTIERPF' );  select \* from all\_sequences where sequence\_owner='VM1DTA' and sequence\_name in (  'SEQ\_GCHPPF',  'SEQ\_GMHDPF',  'SEQ\_GMHIPF',  'SEQ\_GXHIPF',  'SEQ\_GAPHPF',  'SEQ\_GMOVPF',  'SEQ\_ZCLEPF',  'SEQ\_CHDRPF',  'SEQ\_GCHIPF',  'SEQ\_GPSUPF'  );  insert into vm1dta.Dmbmonpf (BATCH\_NAME,START\_TIME,END\_TIME) VALUES ('MBRIND3',16,30);  select \* from vm1dta.DMBMONPF ;  ALTER SEQUENCE VM1DTA.SEQ\_GCHPPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GMHDPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GMHIPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GXHIPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GCHPPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GMHDPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GMHIPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GXHIPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GAPHPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GMOVPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_ZCLEPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_CHDRPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GCHIPF CACHE 20;  ALTER SEQUENCE VM1DTA.SEQ\_GPSUPF CACHE 20;  select \* from VM1DTA.GCHD;  select \* from TITDMGMBRINDp1@DATABASE\_LINK1;  select \* from VM1DTA.zdoemb00 where ZFILENME='TITDMGMBRINDP1' and INDIC ='S' ;  SELECT \*  FROM TITDMGMBRINDP1@DATABASE\_LINK1  where substr(refnum, 1, 8) IN ('00680125','00680435');  select \* from VM1DTA.ZDCLPF\_1412;--10770418  select \* from VM1DTA.ZDCLPF;--10770418  select \* from vm1dta.clntpf where clntnum IN (select zigvalue from vm1dta.ZDCLPF\_1412);  select \* from vm1dta.ZTIERPF;  select \* from TITDMGMBRINDP1\_FREEPLAN@DATABASE\_LINK1;--767363 Rows  select \* from TITDMGMBRINDP1@DATABASE\_LINK1 where refnum like '%13643E44%';-- 288018 Rows  select \* from vm1dta.gchppf where chdrnum IN (select chdrnum from vm1dta.DM2\_PP\_CHDRNUM);  select \* from VM1DTA.ztierpf;  select \* from VM1DTA.DMBMONPF;  select \* from vm1dta.zcelinkpf where clntnum in (  select clntnum from TITDMGMBRINDP1\_FREEPLAN\_CH);  create table TITDMGMBRINDP1\_FREEPLAN\_CH as  select chdrnum , cownnum from vm1dta.chdrpf where chdrnum In ( select SUBSTR(refnum,1,8) from TITDMGMBRINDP1\_FREEPLAN@DATABASE\_LINK1 );  select \* from vm1dta.chdrpf where chdrnum In (select chdrnum from TITDMGMBRINDP1\_CH );  select \* from vm1dta.chdrpf where chdrnum In ('00287024','00287571','00287580');  select \* from vm1dta.chdrpf where chdrnum In (select chdrnum from TITDMGMBRINDP1\_CH );  select \* from vm1dta.ztemptierpf where chdrnum In (select chdrnum from TITDMGMBRINDP1\_CH );  UPDATE vm1dta.titdmgmbrindp2\_P t1  SET (t1.RECIDXMBINDP3) = (  SELECT t2.RECIDXMBINP1  FROM TITDMGMBRINDP1@database\_link1 t2  WHERE t1.refnum = t2.refnum)  WHERE EXISTS (  SELECT 1  FROM TITDMGMBRINDP1@database\_link1 t2  WHERE t1.refnum = t2.refnum );  select count(1) from all\_tables where owner ='VM1DTA' and table\_name = 'ZDOEMB0049';  SELECT \*  FROM user\_parallel\_execute\_chunks  WHERE task\_name = 'DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK'  ORDER BY chunk\_id;  SELECT status, count(1)  FROM user\_parallel\_execute\_chunks  WHERE task\_name = 'DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK'  group by status;  select \* FROM user\_parallel\_execute\_tasks WHERE task\_name = 'DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK';  select \* from DATABASE\_PROPERTIES ;  SHOW PARAMETERS ;  select \* from vm1dta.titdmgmbrindp2\_P order by RECIDXMBINDP3 asc ;  select \* from vm1dta.titdmgmbrindp1\_P;  begin  DBMS\_PARALLEL\_EXECUTE.DROP\_TASK('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK');  DBMS\_PARALLEL\_EXECUTE.CREATE\_TASK('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK');  DBMS\_PARALLEL\_EXECUTE.CREATE\_CHUNKS\_BY\_NUMBER\_COL('DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK','VM1DTA','TITDMGMBRINDP1\_P','RECIDXMBINP1',1000);  end;  /  Delete from vm1dta.titdmgmbrindp1\_p where RECIDXMBINP1 > 1759650;  select 1749650+10000 from dual;  select \* from vm1dta.ZDOEMB0060;  select \* from vm1dta.dmlog;  select \* from VM1DTA.dmbmonpf where JOB\_NUM='60';--3567781  select 3567781/1000 from dual;  create table Vm1dta.ZDOEIN0069 as select \* from Vm1dta.ZDOEPF;  select \* from vm1dta.chdrpf where chdrnum In (select chdrnum from TITDMGMBRINDP1\_CH );    SELECT \*  FROM vm1dta.titdmgmbrindp1\_P;  WHERE RECIDXMBINP1 between 1749652 and 1749653  ORDER BY LPAD(refnum, 15, '0') ASC;--freeplan    truncate table vm1dta.titdmgmbrindp1\_P;  INSERT /\*+ APPEND \*/ INTO vm1dta.titdmgmbrindp1\_P SELECT \* FROM titdmgmbrindp1@database\_link1;  select \* from vm1dta.zdoemb0066;  SELECT \* FROM V$PX\_PROCESS;  SELECT NAME, VALUE FROM GV$SYSSTAT  WHERE UPPER (NAME) LIKE '%PARALLEL OPERATIONS%'  OR UPPER (NAME) LIKE '%PARALLELIZED%' OR UPPER (NAME) LIKE '%PX%';  sELECT status , count(1)  FROM user\_parallel\_execute\_chunks where task\_name ='UPDATE\_CLNTPF\_MERGE'  group by status;  sELECT \*  FROM user\_parallel\_execute\_chunks where task\_name ='DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK';  select \* from vm1dta.ZDRPPF;  select \* from V$PGASTAT;  select resource\_name, current\_utilization, max\_utilization from v$resource\_limit where resource\_name in ('processes','sessions');  select sql\_text, count(\*) as "OPEN CURSORS", user\_name from v$open\_cursor group by sql\_text, user\_name order by count(\*) desc;  select \* from vm1dta.CLNTPF;  SELECT \*  FROM user\_scheduler\_job\_run\_details  WHERE job\_name LIKE (SELECT job\_prefix || '%'  FROM user\_parallel\_execute\_tasks  WHERE task\_name = 'DM2\_MBRIND\_PARALLEL\_EXECUTE\_TASK');  select \* from vm1dta.zdoemb0062;  truncate table vm1dta.zdoeMB0063 ;  truncate table vm1dta.zdoeMB0068;  select \* from vm1dta.dmbmonpf;  select 265632/100 from dual;  select count(1) from VM1DTA.gaphpf where chdrnum In (select chdrnum from TITDMGMBRINDP1\_CH );  select \* from vm1dta.gxhipf where chdrnum In (select chdrnum from TITDMGMBRINDP1\_CH );  SELECT A.\*, B.CHDRNUM AS CHDRNUM\_B  FROM vm1dta.titdmgmbrindp2\_P A  LEFT OUTER JOIN VM1DTA.ZDRPPF B  ON SUBSTR(A.refnum, 1, 8) = B.CHDRNUM  WHERE A.RECIDXMBINDP3 between 1749650 and 1790649  ORDER BY LPAD(A.refnum, 15, '0') ASC;    select \* from vm1dta.titdmgmbrindp2\_P;  select \* from vm1dta.titdmgmbrindp1\_p;    UPDATE vm1dta.titdmgmbrindp2\_P t1  SET (t1.clnt) = (  SELECT t2.RECIDXMBINP1  FROM TITDMGMBRINDP1@database\_link1 t2  WHERE t1.refnum = t2.refnum)  WHERE EXISTS (  SELECT 1  FROM TITDMGMBRINDP1@database\_link1 t2  WHERE t1.refnum = t2.refnum );        select \* from vm1dta.Dmbmonpf;  select \* from vm1dta.zdoemb0069;  select \* from vm1dta.zdoein0069;  select \* from TITDMGMBRINDP1\_CH;  select \* from all\_tables where table\_name like '%ZDOE%'; |