SAS Project Codes

libname raps ORACLE PATH = ODSPROD USER ="xxxx" PW ="xxxx" schema=RAPS ; **run**;

libname REVREC sasspds 'revrec' user=" xxxx " password=" xxxx " server=localhost.5410;

libname DXHCC "/data/dept/ACT/RAPS/RAPS Submissions/DXHCC";

libname DXHCCSTG "/data/dept/ACT/RAPS/RAPS Submissions/DXHCCSTG";

LIBNAME ERPROD SQLSVR USER="SASUser" PASSWORD="sa5us3r8" DATASRC='ERDBPROD\_ENCREP' SCHEMA=ERSVC;

libname RAPSEDPS '/data/dept/ACT/RAPS/RAPS Submissions/RAPSEDPS';

LIBNAME ENC SQLSVR USER="SASUser" PASSWORD=" xxxx" DATASRC='ERDBPROD\_ENCREP' SCHEMA=ERSVC;

/\*import the V22\_HIERARCHIES.CSV as a rapsedps.v22\_hierarchies\*/

/\*data rapsedps.V22\_HIERARCHIES;\*/

/\*SET V22\_HIERARCHIES;\*/

/\*RUN;\*/

/\*Import the chart\_review\_delete\_icn\_and\_cvk\_found.csv to RAPSEDPS.chart\_review\_delete\_icn\_found\*/

/\*data rapsedps.CHART\_REVIEW\_DELETE\_ICN\_AND\_CVK\_;\*/

/\*set CHART\_REVIEW\_DELETE\_ICN\_AND\_CVK\_;\*/

/\*run;\*/

/\* Member Universe - Medicare members with member month >=1 and payment month >=1

from revrec.member\_reporting \_intell with crosswalk of MBI\_XWALK

first if MBI exist, then look for medicare and old medicare \*/

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_test AS

SELECT COALESCE(t5.MBI,t4.MBI,T1.MEDICARE\_NO) AS MBI,

t1.seq\_memb\_id,

t1.subscriber\_id,

T1.GENDER,

T1.DOB,

T1.MBR\_LAST\_NAME,

T1.MBR\_FIRST\_NAME,

t1.LINE\_OF\_BUSINESS AS LOB,

t2.CONTRACT\_NUMBER AS HPLAN,

t1.PY\_STATE,

t1.ACTIVE\_STATUS,

t1.PAYMENT\_MONTHS,

t1.DOS\_YEAR,

t1.IPA\_ID,

t1.IPA\_NAME,

t1.IPAA,

t1.IPAA\_NAME,

t1.PROVIDER\_ID,

t1.PROVIDER\_NPI,

t1.PROV\_FIRST\_NAME,

t1.PROV\_LAST\_NAME

FROM REVREC.MEMBER\_REPORTING\_INTELL t1

LEFT JOIN DXHCC.CONTRACT\_SUBMITTER T2 ON T1.LINE\_OF\_BUSINESS=T2.LOB

LEFT JOIN REVREC.LKP\_MBI\_XWALK t4 ON (t1.MEDICARE\_NO = t4.HICN)

LEFT JOIN REVREC.LKP\_MBI\_XWALK t5 ON (t1.MEDICARE\_NO = t5.MBI)

WHERE t1.PAYMENT\_YEAR = **2019** AND t1.DOS\_YEAR = **2018** AND t1.PAYMENT\_MONTHS > **0** AND t1.DOS\_MONTHS > **0** and t1.LINE\_OF\_BUSINESS NOT IN ('NA','UAR');

**QUIT**;

**proc** **sql**;

create table by\_ipa as

select distinct IPA\_ID,IPA\_NAME,IPAA,IPAA\_NAME,COUNT(MBI)AS MBI\_CNT

FROM RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_test

GROUP BY IPA\_ID,IPA\_NAME,IPAA,IPAA\_NAME;

**QUIT**;

/\* extract RAPS Universe from REVREC.GJD\_RAPS\_RAW and flag RAPS\_IP where type= 01,02 (inpatient) \*/

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.RAPS\_UNIVERSE\_PY2019\_test1 AS

SELECT t1.HICN,

t1.SOURCE,

t1.SUB\_SOURCE,

t1.DATA\_YEAR,

case when t1.dx in ('F23','F28','F29','F531') then **9999** else t2.HCC end AS V22\_HCC,

CASE WHEN t1.TYPE IN ('01','02') THEN 'Y' ELSE 'N' END AS RAP\_IP

FROM REVREC.GJD\_RAPS\_RAW t1

LEFT JOIN DXHCC.V22\_HCC\_NEW T2 ON T1.DX=T2.DX

WHERE t1.DATA\_YEAR = **2018**;

**QUIT**;

**PROC** **SORT** DATA=RAPSEDPS.RAPS\_UNIVERSE\_PY2019\_test1 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.RAPS\_UNIVERSE\_HCC\_test1 AS

SELECT t1.HICN,

COALESCE(t2.MBI,t3.MBI,t1.HICN) AS MBI,

t1.DATA\_YEAR AS DOS\_YEAR ,

t1.V22\_HCC

FROM RAPSEDPS.RAPS\_UNIVERSE\_PY2019\_test1 t1

LEFT JOIN REVREC.LKP\_MBI\_XWALK t2 ON (t1.HICN = t2.HICN)

LEFT JOIN REVREC.LKP\_MBI\_XWALK t3 ON (t1.HICN = t3.MBI);

**QUIT**;

**PROC** **SORT** DATA=RAPSEDPS.RAPS\_UNIVERSE\_HCC\_test1 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE GET\_HCC\_LOW AS

SELECT T1.\*,

T2.V22\_HCC\_LOW

FROM RAPSEDPS.RAPS\_UNIVERSE\_HCC\_test1 T1

LEFT JOIN RAPSEDPS.V22\_HIERARCHIES T2 ON

T1.V22\_HCC=T2.V22\_HCC\_HIGH WHERE T2.V22\_HCC\_LOW NOT IS MISSING;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.RAPS\_UNIVERSE\_HCC\_FINAL\_test1 AS

SELECT T1.\*,

t3.V22\_HCC\_High

FROM RAPSEDPS.RAPS\_UNIVERSE\_HCC\_test1 T1

LEFT JOIN GET\_HCC\_LOW T2 ON T1.MBI=T2.MBI AND T1.V22\_HCC=T2.V22\_HCC\_LOW

LEFT JOIN RAPSEDPS.V22\_HIERARCHIES t3 ON t1.V22\_HCC=t3.V22\_HCC\_LOW

WHERE T2.V22\_HCC\_LOW is missing;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE V23\_RAPS\_IP AS

SELECT t1.HICN,

t1.DATA\_YEAR,

t2.HCC AS V23\_HCC

FROM REVREC.GJD\_RAPS\_RAW t1

INNER JOIN DXHCC.V23\_HCC\_NEW T2 ON T1.DX=T2.DX

WHERE t1.DATA\_YEAR = **2018** AND t1.TYPE IN ('01','02');

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.RAPS\_IP\_HCC\_test1 AS

SELECT t1.HICN,

COALESCE(t2.MBI,t3.MBI,t1.HICN) AS MBI,

t1.DATA\_YEAR AS DOS\_YEAR,

t1.V23\_HCC

FROM V23\_RAPS\_IP t1

LEFT JOIN REVREC.LKP\_MBI\_XWALK t2 ON (t1.HICN = t2.MBI)

LEFT JOIN REVREC.LKP\_MBI\_XWALK t3 ON (t1.HICN = t3.HICN);

**QUIT**;

**PROC** **SORT** DATA=RAPSEDPS.RAPS\_IP\_HCC\_test1 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.ER\_UNIVERSE\_HCC\_test\_2 AS

SELECT t1.MEDICARE\_NO as HICN,

COALESCE(t2.MBI,t3.MBI,t1.MEDICARE\_NO) AS MBI,

t1.seq\_memb\_id,

t1.claim\_key,

t1.claim\_version\_key,

t1.frequency\_code,

t1.version\_no,

t1.DOS\_YEAR,

t1.patient\_account\_number,

t1.data\_source,

T1.DIAGNOSIS\_CODE,

t1.last\_modified\_date,

t1.mao004,

t4.HCC AS V23\_HCC

FROM REVREC.GJD\_MEDICARE\_CLAIMS t1

LEFT JOIN REVREC.LKP\_MBI\_XWALK t2 ON (t1.MEDICARE\_NO = t2.MBI)

LEFT JOIN REVREC.LKP\_MBI\_XWALK t3 ON (t1.MEDICARE\_NO = t3.HICN)

LEFT JOIN DXHCC.V23\_HCC\_NEW T4 ON T1.DIAGNOSIS\_CODE=T4.DX

where t1.DOS\_YEAR = **2018** and t1.ENCOUNTER\_STATUS\_KEY IN (**602**,**60**,**618**,**617**) AND t1.MEDICARE\_NO NOT IS MISSING AND t4.HCC NOT IS MISSING

and T1.MAO004='Y'

AND T1.claim\_version\_key NOT IN (SELECT CLAIM\_VERSION\_KEY FROM rapsedps.CHART\_REVIEW\_DELETE\_ICN\_AND\_CVK\_)

/\* to remove the ck, cvk of the imported chart\_review\_delete\_icn and cvk found 08122019\*/;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.ER\_UNIVERSE\_HCC\_test\_2\_b AS

SELECT t5.OLD\_HICN as HICN,

COALESCE(t5.NEW\_HICN,t1.MEDICARE\_NO) AS MBI,

t1.seq\_memb\_id,

t1.claim\_key,

t1.claim\_version\_key,

t1.encounter\_status\_key as esk,

t1.frequency\_code,

t1.version\_no,

t1.DOS\_YEAR,

t1.patient\_account\_number,

t1.data\_source,

t1.last\_modified\_date,

t1.mao004,

T1.DIAGNOSIS\_CODE,

t4.HCC AS V23\_HCC

FROM REVREC.GJD\_MEDICARE\_CLAIMS t1

INNER JOIN DXHCC.MEMBER\_CROSSWALK t5 on (t1.seq\_memb\_id = t5.seq\_memb\_id)

LEFT JOIN DXHCC.V23\_HCC\_NEW T4 ON T1.DIAGNOSIS\_CODE=T4.DX

where t1.DOS\_YEAR = **2018** and t1.ENCOUNTER\_STATUS\_KEY IN (**602**,**60**,**618**,**617**) AND

t1.MEDICARE\_NO IS MISSING AND t4.HCC NOT IS MISSING and T1.MAO004='Y'

AND T1.claim\_version\_key NOT IN (SELECT CLAIM\_VERSION\_KEY FROM RAPSEDPS.CHART\_REVIEW\_DELETE\_ICN\_AND\_CVK\_)

/\* to remove the ck, cvk of the imported chart\_review\_delete\_icn and cvk found 08122019\*/;

**QUIT**;

**DATA** ER\_UNIVERSE\_HCC\_test\_FINAL;

SET RAPSEDPS.ER\_UNIVERSE\_HCC\_test\_2\_b RAPSEDPS.ER\_UNIVERSE\_HCC\_test\_2;

**RUN**;

**PROC** **SORT** DATA=ER\_UNIVERSE\_HCC\_test\_FINAL NODUPKEY; BY \_ALL\_;**RUN**;

**proc** **sql**;

create table MAX\_DETAIL as

select a.claim\_key, max(a.claim\_version\_key) as MAX\_CVK

from ER\_UNIVERSE\_HCC\_test\_FINAL a

group by a.claim\_key;

**run**;

**proc** **sql**;

create table RAPSEDPS.er\_universe\_hcc\_test\_3 as

select a.\*

from ER\_UNIVERSE\_HCC\_test\_FINAL a

join work.max\_detail b

on a.claim\_version\_key = b.max\_cvk

where esk not in (**84**,**74**)

and frequency\_code not in ('8');

**run**;

**PROC** **SORT** DATA=RAPSEDPS.er\_universe\_hcc\_test\_3 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **APPEND** BASE=RAPSEDPS.er\_universe\_hcc\_test\_3 DATA=RAPSEDPS.RAPS\_IP\_HCC\_test1 FORCE;**RUN**;

**PROC** **SORT** DATA=RAPSEDPS.er\_universe\_hcc\_test\_3 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE GET\_HCC\_LOW\_ER AS

SELECT T1.\*,

T2.V23\_HCC\_LOW

FROM RAPSEDPS.er\_universe\_hcc\_test\_3 T1

LEFT JOIN DXHCC.V23\_HIERARCHIES T2 ON

T1.V23\_HCC=T2.V23\_HCC\_HIGH WHERE T2.V23\_HCC\_LOW NOT IS MISSING;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.ER\_UNIVERSE\_HCC\_FINAL\_test\_2 AS

SELECT T1.\*,

t3.V23\_HCC\_High ,CASE WHEN T1.V23\_HCC IN (**56**,**59**,**60**,**138**) THEN 'Y' ELSE 'N' END AS ENC\_HCC\_ONLY

FROM RAPSEDPS.er\_universe\_hcc\_test\_3 T1

LEFT JOIN GET\_HCC\_LOW\_ER T2 ON T1.MBI=T2.MBI AND T1.V23\_HCC=T2.V23\_HCC\_LOW

LEFT JOIN DXHCC.V23\_HIERARCHIES t3 ON t1.V23\_HCC=t3.V23\_HCC\_LOW

WHERE T2.V23\_HCC\_LOW is missing;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.MAO004\_UNIVERSE\_HCC\_test\_2 AS

SELECT t1.MEDICARE\_NO as HICN,

COALESCE(t2.MBI,t3.MBI,t1.MEDICARE\_NO) AS MBI,

t1.DOS\_YEAR,

t2.HCC AS V23\_HCC

FROM RAPSEDPS.MAO\_004\_PHASE3\_ALLOW t1

INNER JOIN DXHCC.V23\_HCC\_NEW T2 ON T1.DIAGNOSIS\_CODE=T2.DX

LEFT JOIN REVREC.LKP\_MBI\_XWALK t2 ON (t1.MEDICARE\_NO = t2.HICN)

LEFT JOIN REVREC.LKP\_MBI\_XWALK t3 ON (t1.MEDICARE\_NO = t3.MBI)

WHERE t1.A\_D\_FLAG = 'A' AND t1.ADD\_OR\_DELETE = 'A' AND t1.DOS\_YEAR = **2018**

AND T1.ICN NOT IN (SELECT REPORTING\_ENTITY\_CONTROL\_NUMBER FROM rapsedps.CHART\_REVIEW\_DELETE\_ICN\_AND\_CVK\_)

/\* to remove the ck, cvk of the imported chart\_reiew\_delete\_icn and cvk found 08122019\*/

;

**QUIT**;

**PROC** **SORT** DATA=RAPSEDPS.MAO004\_UNIVERSE\_HCC\_test\_2 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **APPEND** BASE=RAPSEDPS.MAO004\_UNIVERSE\_HCC\_test\_2 DATA=RAPSEDPS.RAPS\_IP\_HCC\_test1 FORCE;**RUN**;

**PROC** **SORT** DATA=RAPSEDPS.MAO004\_UNIVERSE\_HCC\_test\_2 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE GET\_HCC\_LOW\_MAO AS

SELECT T1.\*,

T2.V23\_HCC\_LOW

FROM RAPSEDPS.MAO004\_UNIVERSE\_HCC\_test\_2 T1

LEFT JOIN DXHCC.V23\_HIERARCHIES T2 ON

T1.V23\_HCC=T2.V23\_HCC\_HIGH WHERE T2.V23\_HCC\_LOW NOT IS MISSING

;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.MAO004\_UNIVERSE\_HCC\_FINAL\_test\_2 AS

SELECT T1.\*,

t3.V23\_HCC\_High ,CASE WHEN T1.V23\_HCC IN (**56**,**59**,**60**,**138**) THEN 'Y' ELSE 'N' END AS ENC\_HCC\_ONLY

FROM RAPSEDPS.MAO004\_UNIVERSE\_HCC\_test\_2 T1

LEFT JOIN GET\_HCC\_LOW\_MAO T2 ON T1.MBI=T2.MBI AND T1.V23\_HCC=T2.V23\_HCC\_LOW

LEFT JOIN DXHCC.V23\_HIERARCHIES t3 ON t1.V23\_HCC=t3.V23\_HCC\_LOW

WHERE T2.V23\_HCC\_LOW is missing

;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE TEST AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

CASE WHEN t1.V22\_HCC=**58** THEN **59**

WHEN t1.V22\_HCC=**9999** THEN **58** ELSE t1.V22\_HCC END AS HCC

/\*make sure that 58 is still there for the F codes\*/

FROM RAPSEDPS.RAPS\_UNIVERSE\_HCC\_test1 t1;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.RAPS\_HCC\_UNI\_test\_2 AS

SELECT DISTINCT t1.\*,

T2.AVG\_SCORE

FROM TEST t1

LEFT JOIN DXHCC.V22\_V23\_AVG\_SCORES T2 ON T1.HCC=T2.HCC

WHERE t1.hcc is not missing

;**QUIT**;

**DATA** EDPSHCCCOMBINED;

SET RAPSEDPS.MAO004\_UNIVERSE\_HCC\_test\_2 RAPSEDPS.ER\_UNIVERSE\_HCC\_FINAL\_test\_2;

**RUN**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.EDPS\_HCC\_UNI\_test\_2 AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.V23\_HCC AS HCC,

T2.AVG\_SCORE

FROM EDPSHCCCOMBINED t1

LEFT JOIN DXHCC.V22\_V23\_AVG\_SCORES T2 ON T1.V23\_HCC=T2.HCC;

**QUIT**;

**DATA** RAPSEDPS.RAPSEDPS\_HCC\_UNIVERSE\_test\_1;

SET RAPSEDPS.RAPS\_HCC\_UNI\_test\_2 RAPSEDPS.EDPS\_HCC\_UNI\_test\_2;

**RUN**;

**PROC** **SORT** DATA=RAPSEDPS.RAPSEDPS\_HCC\_UNIVERSE\_test\_1 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_HCC\_test1 AS

SELECT DISTINCT t1.MBI,

t1.LOB,

t1.HPLAN,

t1.PY\_STATE,

t1.ACTIVE\_STATUS,

t1.PAYMENT\_MONTHS,

t2.DOS\_YEAR,

t2.HCC,

T2.AVG\_SCORE

FROM RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_test t1

JOIN RAPSEDPS.RAPSEDPS\_HCC\_UNIVERSE\_test\_1 t2 ON (t1.MBI = t2.MBI);

**QUIT**;

**PROC** **SQL**;

CREATE TABLE GET\_HCC\_LOW\_ALL AS

SELECT T1.\*,

T2.V22\_HCC\_LOW

FROM RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_HCC\_test1 T1

LEFT JOIN RAPSEDPS.V22\_HIERARCHIES T2 ON T1.HCC=T2.V22\_HCC\_HIGH WHERE T2.V22\_HCC\_LOW NOT IS MISSING;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE HCC\_UNI\_ALL AS

SELECT T1.\*,

t3.V22\_HCC\_High AS HCC\_HIGH, T4.V22\_HCC\_LOW AS HCC\_LOW

FROM RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_HCC\_test1 T1

LEFT JOIN GET\_HCC\_LOW\_ALL T2 ON T1.MBI=T2.MBI AND T1.HCC=T2.V22\_HCC\_LOW

LEFT JOIN RAPSEDPS.V22\_HIERARCHIES t3 ON t1.HCC=t3.V22\_HCC\_LOW

LEFT JOIN RAPSEDPS.V22\_HIERARCHIES t4 ON t1.HCC=t4.V22\_HCC\_HIGH

WHERE T2.V22\_HCC\_LOW is missing;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.MEMBER\_UNIV\_PY2019\_HCC\_FINAL1 AS

SELECT DISTINCT t1.MBI,

t1.LOB,

t1.HPLAN,

t1.PY\_STATE,

t1.ACTIVE\_STATUS,

t1.PAYMENT\_MONTHS,

t1.DOS\_YEAR,

t1.HCC,

t1.AVG\_SCORE AS HCC\_SCORE,

t1.HCC\_HIGH,

t2.AVG\_SCORE AS HCC\_SCORE\_HIGH,

t1.HCC\_LOW,

t3.AVG\_SCORE AS HCC\_SCORE\_LOW

FROM HCC\_UNI\_ALL t1

LEFT JOIN DXHCC.V22\_V23\_AVG\_SCORES t2 ON (t1.HCC\_HIGH = t2.HCC)

LEFT JOIN DXHCC.V22\_V23\_AVG\_SCORES t3 ON (t1.HCC\_LOW = t3.HCC);

**QUIT**;

**PROC** **SQL**;

CREATE TABLE HCC\_IN\_V23 AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.HCC,

t1.HCC\_SCORE

FROM RAPSEDPS.MEMBER\_UNIV\_PY2019\_HCC\_FINAL1 t1

WHERE T1.HCC IN (**56**,**59**,**60**,**138**);

**QUIT**;

**PROC** **SQL**;

CREATE TABLE HCC\_IN\_BOTH AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.HCC,

t1.HCC\_SCORE,

T2.HCC as RAPS,

T3.HCC AS ENC

FROM RAPSEDPS.MEMBER\_UNIV\_PY2019\_HCC\_FINAL1 t1

LEFT JOIN RAPSEDPS.RAPS\_HCC\_UNI\_test\_2 t2 ON (t1.MBI = t2.MBI) AND (t1.DOS\_YEAR = t2.DOS\_YEAR) AND (t1.HCC = t2.HCC)

LEFT JOIN RAPSEDPS.EDPS\_HCC\_UNI\_test\_2 t3 ON (t1.MBI = t3.MBI) AND (t1.DOS\_YEAR = t3.DOS\_YEAR) AND (t1.HCC = t3.HCC)

WHERE T2.HCC NOT IS MISSING AND T3.HCC NOT IS MISSING

;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE HCC\_IN\_RAPS AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.HCC,

t1.HCC\_SCORE,

CASE WHEN t6.HCC NOT IS MISSING THEN 'N'

WHEN t6.HCC IS MISSING and t7.HCC NOT IS MISSING AND T1.HCC\_SCORE <= T7.AVG\_SCORE THEN 'N'

ELSE 'Y' END AS GAP

FROM RAPSEDPS.MEMBER\_UNIV\_PY2019\_HCC\_FINAL1 t1

LEFT JOIN RAPSEDPS.RAPS\_HCC\_UNI\_test\_2 t2 ON (t1.MBI = t2.MBI) AND (t1.DOS\_YEAR = t2.DOS\_YEAR) AND (t1.HCC = t2.HCC)

LEFT JOIN RAPSEDPS.EDPS\_HCC\_UNI\_test\_2 t3 ON (t1.MBI = t3.MBI) AND (t1.DOS\_YEAR = t3.DOS\_YEAR) AND (t1.HCC = t3.HCC)

LEFT JOIN HCC\_IN\_BOTH t4 ON (t1.MBI = t4.MBI) AND (t1.DOS\_YEAR = t4.DOS\_YEAR) AND (t1.HCC = t4.HCC)

LEFT JOIN HCC\_IN\_V23 T5 ON (t1.MBI = T5.MBI) AND (t1.DOS\_YEAR = T5.DOS\_YEAR) AND (t1.HCC = T5.HCC)

/\*Check higher and lower HCC match\*/

LEFT JOIN RAPSEDPS.EDPS\_HCC\_UNI\_test\_2 t6 ON (t1.MBI = t6.MBI) AND (t1.DOS\_YEAR = t6.DOS\_YEAR) AND (t1.HCC\_HIGH = t6.HCC) /\*EDPS HIGHER\*/

LEFT JOIN RAPSEDPS.EDPS\_HCC\_UNI\_test\_2 t7 ON (t1.MBI = t7.MBI) AND (t1.DOS\_YEAR = t7.DOS\_YEAR) AND (t1.HCC\_LOW = t7.HCC) /\*EDPS LOWER\*/

WHERE T2.HCC NOT IS MISSING AND T3.HCC IS MISSING AND T4.HCC IS MISSING AND T5.HCC IS MISSING;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE HCC\_IN\_ENC AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.HCC,

t1.HCC\_SCORE,

CASE WHEN t6.HCC NOT IS MISSING THEN 'N'

WHEN t6.HCC IS MISSING and t7.HCC NOT IS MISSING AND T1.HCC\_SCORE <= T7.AVG\_SCORE THEN 'N'

ELSE 'Y' END AS GAP

FROM RAPSEDPS.MEMBER\_UNIV\_PY2019\_HCC\_FINAL1 t1

LEFT JOIN RAPSEDPS.RAPS\_HCC\_UNI\_test\_2 t2 ON (t1.MBI = t2.MBI) AND (t1.DOS\_YEAR = t2.DOS\_YEAR) AND (t1.HCC = t2.HCC)

LEFT JOIN RAPSEDPS.EDPS\_HCC\_UNI\_test\_2 t3 ON (t1.MBI = t3.MBI) AND (t1.DOS\_YEAR = t3.DOS\_YEAR) AND (t1.HCC = t3.HCC)

LEFT JOIN HCC\_IN\_BOTH t4 ON (t1.MBI = t4.MBI) AND (t1.DOS\_YEAR = t4.DOS\_YEAR) AND (t1.HCC = t4.HCC)

LEFT JOIN HCC\_IN\_V23 T5 ON (t1.MBI = T5.MBI) AND (t1.DOS\_YEAR = T5.DOS\_YEAR) AND (t1.HCC = T5.HCC)

/\*Check higher and lower HCC match\*/

LEFT JOIN RAPSEDPS.RAPS\_HCC\_UNI\_test\_2 t6 ON (t1.MBI = t6.MBI) AND (t1.DOS\_YEAR = t6.DOS\_YEAR) AND (t1.HCC\_HIGH = t6.HCC) /\*RAPS HIGHER\*/

LEFT JOIN RAPSEDPS.RAPS\_HCC\_UNI\_test\_2 t7 ON (t1.MBI = t7.MBI) AND (t1.DOS\_YEAR = t7.DOS\_YEAR) AND (t1.HCC\_LOW = t7.HCC) /\*RAPS LOWER\*/

WHERE T2.HCC IS MISSING AND T3.HCC NOT IS MISSING AND T4.HCC IS MISSING AND T5.HCC IS MISSING;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE HIGH\_LOW\_FOUND\_ENC AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.HCC,

t1.HCC\_SCORE

FROM HCC\_IN\_ENC t1

WHERE t1.GAP = 'N';

**QUIT**;

**PROC** **SQL**;

CREATE TABLE HIGH\_LOW\_FOUND\_RAPS AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.HCC,

t1.HCC\_SCORE

FROM HCC\_IN\_RAPS t1

WHERE t1.GAP = 'N';

**QUIT**;

**DATA** HCC\_IN\_BOTH\_ADDTL;

SET HIGH\_LOW\_FOUND\_RAPS HIGH\_LOW\_FOUND\_ENC;

**RUN**;

**PROC** **SORT** DATA=HCC\_IN\_BOTH\_ADDTL NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **APPEND** BASE=HCC\_IN\_BOTH DATA=HCC\_IN\_BOTH\_ADDTL FORCE; **RUN**;

**PROC** **SQL**;

CREATE TABLE IN\_ENC AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.HCC,

t1.HCC\_SCORE

FROM HCC\_IN\_ENC t1

WHERE t1.GAP = 'Y';

**QUIT**;

**PROC** **SQL**;

CREATE TABLE IN\_RAPS AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.HCC,

t1.HCC\_SCORE

FROM HCC\_IN\_RAPS t1

WHERE t1.GAP = 'Y';

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS\_PY2019\_HCC\_RECON\_FLAG AS

SELECT DISTINCT t1.MBI,

t1.LOB,

t1.HPLAN,

t1.PY\_STATE,

t1.ACTIVE\_STATUS,

t1.PAYMENT\_MONTHS,

t1.DOS\_YEAR,

t1.HCC, t2.MBI as BOTH, t3.MBI AS RAPS, t4.MBI AS ENC, t5.MBI AS V23ONLY,

CASE WHEN t2.MBI NOT IS MISSING THEN 'HCC IN RAPS & ENCOUNTERS'

WHEN t3.MBI NOT IS MISSING THEN 'HCC IN RAPS NOT IN ENCOUNTERS'

WHEN t4.MBI NOT IS MISSING THEN 'HCC IN ENCOUNTERS NOT IN RAPS'

WHEN t5.MBI NOT IS MISSING THEN 'HCC IN ENCOUNTERS V23 ONLY ' END AS HCC\_Bucket

FROM RAPSEDPS.MEMBER\_UNIV\_PY2019\_HCC\_FINAL1 t1

LEFT JOIN WORK.HCC\_IN\_BOTH t2 ON (t1.MBI = t2.MBI) AND (t1.DOS\_YEAR = t2.DOS\_YEAR) AND (t1.HCC = t2.HCC)

LEFT JOIN WORK.IN\_RAPS t3 ON (t1.MBI = t3.MBI) AND (t1.DOS\_YEAR = t3.DOS\_YEAR) AND (t1.HCC = t3.HCC)

LEFT JOIN WORK.IN\_ENC t4 ON (t1.MBI = t4.MBI) AND (t1.DOS\_YEAR = t4.DOS\_YEAR) AND (t1.HCC = t4.HCC)

LEFT JOIN WORK.HCC\_IN\_V23 t5 ON (t1.MBI = t5.MBI) AND (t1.DOS\_YEAR = t5.DOS\_YEAR) AND (t1.HCC = t5.HCC);

**QUIT**;

**DATA** RAPSEDPS.RAPSEDPS\_PY2019\_HCC\_RECON\_FINALA;

SET RAPSEDPS\_PY2019\_HCC\_RECON\_FLAG;

IF HCC\_Bucket='HCC IN RAPS & ENCOUNTERS' THEN HCC\_STATUS\_CATEGORY='HCC CLOSED';

IF HCC\_Bucket='HCC IN ENCOUNTERS V23 ONLY' THEN HCC\_STATUS\_CATEGORY='HCC CLOSED';

IF HCC\_Bucket='HCC IN RAPS NOT IN ENCOUNTERS' THEN HCC\_STATUS\_CATEGORY='HCC GAP';

IF HCC\_Bucket='HCC IN ENCOUNTERS NOT IN RAPS' THEN HCC\_STATUS\_CATEGORY='HCC GAP';

**RUN**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS\_PY2019\_RECON\_SUMM AS

SELECT t1.HCC\_STATUS\_CATEGORY,

t1.HCC\_Bucket,

t1.LOB,

t1.DOS\_YEAR,

t1.PAYMENT\_MONTHS,

(COUNT(t1.MBI)) AS TOTAL\_HCC,

%sysfunc(date()) format=mmddyy10. as REPORT\_DATE

from RAPSEDPS.RAPSEDPS\_PY2019\_HCC\_RECON\_FINALA t1

/\*FROM WORK.RAPSEDPS\_PY2019\_HCC\_RECON\_FLAG t1\*/

GROUP BY t1.HCC\_STATUS\_CATEGORY,

t1.HCC\_Bucket,

t1.LOB,

t1.DOS\_YEAR,

t1.PAYMENT\_MONTHS;

**QUIT**;

**PROC** **SQL**;

select

sum(t1.TOTAL\_HCC) into:TOTAL\_HCC

from RAPSEDPS\_PY2019\_RECON\_SUMM t1;

**QUIT**;

**proc** **export** data=RAPSEDPS\_PY2019\_RECON\_SUMM

outfile="/data/dept/ACT/RAPS/RAPS Submissions/RAPSEDPS/Exception\_Reports/RAPSEDPS\_PY2019\_RECON\_SUMM.csv"

dbms=csv

replace;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* This is for Pam and Susan detail data\*/

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.ER\_UNIVERSE\_HCC\_detail\_1 AS

SELECT t1.MEDICARE\_NO as HICN,

COALESCE(t2.MBI,t3.MBI,t1.MEDICARE\_NO) AS MBI,

t1.seq\_memb\_id,

t1.DOS\_YEAR,

t4.HCC AS V23\_HCC,

t1.claim\_key,

t1.claim\_version\_key,

t1.frequency\_code,

t1.version\_no,

t1.last\_modified\_date,

t1.encounter\_status\_key as esk,

case when t1.ENCOUNTER\_STATUS\_KEY in (**602**,**60**,**84**,**617**,**618**) then 'Y'

else '' end as Accepted,

t1.patient\_account\_number,

t1.data\_source,

T1.DIAGNOSIS\_CODE,

T1.MAO004

FROM REVREC.GJD\_MEDICARE\_CLAIMS t1

LEFT JOIN REVREC.LKP\_MBI\_XWALK t2 ON (t1.MEDICARE\_NO = t2.MBI)

LEFT JOIN REVREC.LKP\_MBI\_XWALK t3 ON (t1.MEDICARE\_NO = t3.HICN)

LEFT JOIN DXHCC.V23\_HCC\_NEW T4 ON T1.DIAGNOSIS\_CODE=T4.DX

where t1.DOS\_YEAR = **2018** /\*and t1.ENCOUNTER\_STATUS\_KEY IN (602,60,618,617)\*/ AND t1.MEDICARE\_NO NOT IS MISSING AND t4.HCC NOT IS MISSING

AND T1.claim\_version\_key NOT IN (SELECT CLAIM\_VERSION\_KEY FROM rapsedps.CHART\_REVIEW\_DELETE\_ICN\_AND\_CVK\_)

/\* to remove the ck, cvk of the imported chart\_review\_delete\_icn and cvk found 08122019\*/;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.ER\_UNIVERSE\_HCC\_detail\_2\_b AS

SELECT t5.OLD\_HICN as HICN,

COALESCE(t5.NEW\_HICN,t1.MEDICARE\_NO) AS MBI,

t1.seq\_memb\_id,

t1.claim\_key,

t1.claim\_version\_key,

t1.encounter\_status\_key as esk,

case when t1.ENCOUNTER\_STATUS\_KEY in (**602**,**60**,**84**,**617**,**618**) then 'Y'

else '' end as Accepted,

t1.frequency\_code,

t1.version\_no,

t1.DOS\_YEAR,

t1.patient\_account\_number,

t1.data\_source,

t1.last\_modified\_date,

t1.mao004,

T1.DIAGNOSIS\_CODE,

t4.HCC AS V23\_HCC

FROM REVREC.GJD\_MEDICARE\_CLAIMS t1

INNER JOIN DXHCC.MEMBER\_CROSSWALK t5 on (t1.seq\_memb\_id = t5.seq\_memb\_id)

LEFT JOIN DXHCC.V23\_HCC\_NEW T4 ON T1.DIAGNOSIS\_CODE=T4.DX

where t1.DOS\_YEAR = **2018** /\*and t1.ENCOUNTER\_STATUS\_KEY IN (602,60,618,617)\*/ AND

t1.MEDICARE\_NO IS MISSING AND t4.HCC NOT IS MISSING

AND T1.claim\_version\_key NOT IN (SELECT CLAIM\_VERSION\_KEY FROM rapsedps.CHART\_REVIEW\_DELETE\_ICN\_AND\_CVK\_)

/\* to remove the ck, cvk of the imported chart\_review\_delete\_icn and cvk found 08122019\*/;

**QUIT**;

**DATA** ER\_UNIVERSE\_HCC\_detail\_FINAL;

SET RAPSEDPS.ER\_UNIVERSE\_HCC\_detail\_2\_b RAPSEDPS.ER\_UNIVERSE\_HCC\_detail\_1;

**RUN**;

**PROC** **SORT** DATA=ER\_UNIVERSE\_HCC\_detail\_FINAL NODUPKEY; BY \_ALL\_;**RUN**;

**proc** **sql**;

create table MAX\_DETAIL as

select a.claim\_key, max(a.claim\_version\_key) as MAX\_CVK

from ER\_UNIVERSE\_HCC\_detail\_FINAL a

group by a.claim\_key;

**run**;

**proc** **sql**;

create table RAPSEDPS.er\_universe\_hcc\_detail\_3 as

select a.\*

from ER\_UNIVERSE\_HCC\_detail\_FINAL a

join work.max\_detail b

on a.claim\_version\_key = b.max\_cvk

where esk not in (**84**,**74**)

and frequency\_code not in ('8');

**run**;

**PROC** **SORT** DATA=RAPSEDPS.er\_universe\_hcc\_detail\_3 NODUPKEY; BY \_ALL\_;**RUN**;

**Proc** **sql**;

create table GAP\_IN\_ENC as

select t1.\*

from RAPSEDPS.RAPSEDPS\_PY2019\_HCC\_RECON\_FINALa t1

WHERE T1.HCC\_BUCKET='HCC IN RAPS NOT IN ENCOUNTERS' AND T1.HCC\_STATUS\_CATEGORY='HCC GAP';

**quit**;

**PROC** **SQL**;

CREATE TABLE GAP\_IN\_ENC\_DET AS

SELECT t1.\*,

t2.esk,

MAX(T2.LAST\_MODIFIED\_DATE) AS LAST\_MODIFIED\_DATE FORMAT=MMDDYY10.,

T2.MAO004

FROM GAP\_IN\_ENC t1

LEFT JOIN RAPSEDPS.er\_universe\_hcc\_detail\_3 t2 ON (t1.MBI=t2.MBI) AND T1.DOS\_YEAR=T2.DOS\_YEAR AND T1.HCC=T2.v23\_HCC

GROUP BY t1.MBI,

t1.LOB,

t1.HPLAN,

t1.PY\_STATE,

t1.ACTIVE\_STATUS,

t1.PAYMENT\_MONTHS,

t1.DOS\_YEAR,

t1.HCC,

t1.BOTH,

t1.RAPS,

t1.ENC,

t1.V23ONLY,

t1.HCC\_Bucket,

t1.HCC\_STATUS\_CATEGORY,

t2.esk,

t2.MAO004;

**QUIT**;

**PROC** **SORT** DATA=GAP\_IN\_ENC\_DET NODUPKEY; BY \_ALL\_;**RUN**;

**proc** **sql**;

create table rapsedps.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_a AS

SELECT T1.\*, MAX(T1.LAST\_MODIFIED\_DATE) AS LAST\_MODIFIED\_DATE FORMAT=MMDDYY10.,

t4.Prioritization as Prioritization1

FROM GAP\_IN\_ENC\_DET t1

Left Join RAPSEDPS.NME\_ER\_STATUS\_BUCKETING t4 ON t1.esk = t4.STATUS\_KEY

ORDER BY t1.MBI;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_b AS

SELECT DISTINCT t1.MBI,

t1.HCC, MAX(T1.LAST\_MODIFIED\_DATE) AS LAST\_MODIFIED\_DATE FORMAT=MMDDYY10.,

(MIN(t1.Prioritization1)) FORMAT=BEST12. AS Prioritization,

t1.MAO004

FROM RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_a t1

where t1.MAO004='Y'

group by t1.MBI,

t1.HCC,t1.MAO004;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_c AS

SELECT DISTINCT t1.MBI,

t1.HCC, MAX(T1.LAST\_MODIFIED\_DATE) AS LAST\_MODIFIED\_DATE FORMAT=MMDDYY10.,

(MIN(t1.Prioritization1)) FORMAT=BEST12. AS Prioritization,

t1.MAO004

FROM RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_a t1

left join RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_b t2 on t1.MBI=t2.MBI and t1.HCC=t2.hcc

where t2.mbi is missing

group by t1.MBI,

t1.HCC,t1.MAO004;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_d AS

SELECT \* FROM RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_b

UNION CORRESPONDING

SELECT \* FROM RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_c;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_E AS

SELECT t1.MBI,

t1.HCC,

t1.Prioritization,

t1.MAO004,

T1.LAST\_MODIFIED\_DATE,

t2.Status\_Key AS ESK,

t2.Description,

t2.Processing\_Status

FROM RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_D t1

LEFT JOIN RAPSEDPS.NME\_ER\_STATUS\_BUCKETING t2 ON (t1.Prioritization = t2.Prioritization);

**QUIT**;

/\* get ealiest Claim with that statust \*/

**data** RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_F;

set RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_E ;

today\_date=%sysfunc(date());

format today\_date mmddyy10.;

date\_diff=today\_date-LAST\_MODIFIED\_DATE;

count=**1**;

**run**;

%let today\_date=%sysfunc(date()) format=mmddyy10.;

%put today\_date %putn(&today\_date,mmddyy10.);

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_AGE AS

SELECT t1.Description,t1.Prioritization,

t1.Processing\_Status,t1.ESK, t1.MAO004,sum(t1.count) as concat,

/\* 1-15 DAYS \*/

(Sum(case When t1.date\_diff <= **15** then **1** else **0** End )) AS '1-15 DAYS'n,

/\* 16-30 DAYS \*/

(Sum(case When t1.date\_diff > **15** And t1.date\_diff <= **30** then **1** else **0** End )) AS '16-30 DAYS'n,

/\* 31-60 DAYS \*/

(Sum(case When t1.date\_diff > **30** And t1.date\_diff <= **60** then **1** else **0** End )) AS '31-60 DAYS'n,

/\* 61-120 DAYS \*/

(Sum(case When t1.date\_diff > **60** And t1.date\_diff <= **120** then **1** else **0** End )) AS '61-120 DAYS'n,

/\* OVER 120 DAYS \*/

(Sum(case When t1.date\_diff > **120** then **1** else **0** End )) AS 'OVER 120 DAYS'n

FROM RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_F t1

GROUP BY t1.ESK, t1.MAO004,t1.Description, t1.Prioritization,

t1.Processing\_Status

;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_SUMM AS

SELECT Distinct t1.Prioritization,

t1.Esk as ENCOUNTER\_STATUS\_KEY,

CASE WHEN t1.Esk IS NULL THEN 'NOT IN ER' ELSE t1.Description END AS ER\_STATUS,

CASE WHEN t1.MAO004='Y' THEN 'Passed MAO004' ELSE 'Not Passed MAO004' END AS MAO004,

CASE WHEN t1.MAO004='Y' THEN 'Actionable Gap'

WHEN t1.Processing\_Status IS NULL then 'Actionable Gap' ELSE 'Filtering logic difference' END AS Gap\_Status,

CASE WHEN t1.Esk IS NULL THEN 'HCC Not Found in Encounters - Research required'

WHEN t1.MAO004='Y' THEN T2.REPORTING\_STATUS

WHEN t1.Esk NOT IS NULL AND t1.MAO004='' then 'Filtering logic difference - Added to Chart Chase'

END AS REPORTING\_STATUS,

t1.concat AS TOTAL,

t1.concat/&TOTAL\_HCC. AS PERCENTAGE format **9.4**,

case when t1.Esk IS NULL then **0** else t1.'1-15 DAYS'n end as '1-15 DAYS'n,

t1.'16-30 DAYS'n,

t1.'31-60 DAYS'n,

t1.'61-120 DAYS'n,

t1.'OVER 120 DAYS'n,

%sysfunc(date()) format=mmddyy10. as report\_date,

T2.REPORTING\_STATUS

FROM RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_AGE t1

LEFT JOIN RAPSEDPS.NME\_ER\_STATUS\_BUCKETING t2 ON (t1.Prioritization = t2.Prioritization)

GROUP BY t1.Prioritization,

t1.Esk,

t1.Description,

T1.MAO004,

t1.Processing\_Status, T2.REPORTING\_STATUS;

**QUIT**;

%put &TOTAL\_HCC;

**proc** **export** data=RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_SUMM

outfile="/data/dept/ACT/RAPS/RAPS Submissions/RAPSEDPS/Exception\_Reports/INRAPSONLY\_PY2019\_RECON\_SUMM.csv"

dbms=csv

replace;

**run**;

libname DXHCCSTG "/data/dept/ACT/RAPS/RAPS Submissions/DXHCCSTG";

**proc** **sql**;

create table hcc\_raps\_not\_enc\_er\_status\_1 as

select t1.\*,t2.claim\_key,t2.claim\_version\_key,t2.seq\_memb\_id,

t2.esk,T3.GENDER,

T3.DOB,

T3.MBR\_LAST\_NAME,

T3.MBR\_FIRST\_NAME,

t3.LOB,

MAX(T2.LAST\_MODIFIED\_DATE) AS LAST\_MODIFIED\_DATE FORMAT=MMDDYY10.,

T2.MAO004

from RAPSEDPS.RAPSEDPS\_PY2019\_HCC\_RECON\_FINALa t1

LEFT JOIN RAPSEDPS.er\_universe\_hcc\_detail\_3 t2 ON

(t1.MBI=t2.MBI) AND T1.DOS\_YEAR=T2.DOS\_YEAR AND T1.HCC=T2.v23\_HCC

LEFT JOIN RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_test T3

ON T2.SEQ\_MEMB\_ID=T3.SEQ\_MEMB\_ID

where T1.HCC\_BUCKET='HCC IN RAPS NOT IN ENCOUNTERS' AND T1.HCC\_STATUS\_CATEGORY='HCC GAP' and

t2.esk not is null and t2.mao004='';

**quit**;

**PROC** **SORT** DATA=hcc\_raps\_not\_enc\_er\_status\_1 NODUPKEY; BY \_ALL\_;**RUN**;

/\*need to get the raps\_not\_in\_enc\_hcc\_gap\_v5\_update082319 code and run the "IT and the other" files, to get the

rapsedps.combined\_raps\_not\_in\_enc\*/

**proc** **sql**;

create table CODER\_HCC\_MEMBER\_INFO\_ADJF\_all\_1 as

select t1.\*,t2.claim\_key,t2.claim\_version\_key,t2.seq\_memb\_id,

t2.esk,T3.GENDER,

T3.DOB,

T3.MBR\_LAST\_NAME,

T3.MBR\_FIRST\_NAME,

t3.LOB,

MAX(T2.LAST\_MODIFIED\_DATE) AS LAST\_MODIFIED\_DATE FORMAT=MMDDYY10.,

T2.MAO004

from RAPSEDPS.COMBINED\_RAPS\_NOT\_IN\_ENC t1/\* files for RAPS\_NOT\_IN\_ENC\_HCC\_GAP\_V5\_UPDATE082319\*/

LEFT JOIN RAPSEDPS.er\_universe\_hcc\_detail\_3 t2 ON (t1.MBI=t2.MBI AND T1.HCC=T2.v23\_HCC)

LEFT JOIN RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_test T3

ON T2.SEQ\_MEMB\_ID=T3.SEQ\_MEMB\_ID

where t2.esk not is null and t2.mao004=''

;

**quit**;

**PROC** **SORT** DATA=CODER\_HCC\_MEMBER\_INFO\_ADJF\_all\_1 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE hcc\_raps\_not\_enc\_er\_status\_fINAL AS

SELECT \*

FROM hcc\_raps\_not\_enc\_er\_status\_1

WHERE claim\_Version\_Key not in (select claim\_version\_key from CODER\_HCC\_MEMBER\_INFO\_ADJF\_all\_1);

**quit**;

**proc** **sql**;

create table hcc\_raps\_not\_enc\_er\_status\_fL1 as

select t1.\*,

t2.facility\_type,

t2.place\_of\_service,

t2.diagnosis\_code,

t3.hcc,

t4.procedure\_code

from hcc\_raps\_not\_enc\_er\_status\_fINAL t1 left join revrec.GJD\_MEDICARe\_Claims t2

on t1.claim\_key=t2.claim\_key and t1.claim\_version\_key=t2.claim\_version\_key

left join HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_1 t3

on t1.claim\_key=t3.claim\_key and t1.claim\_version\_key=t3.claim\_version\_key

left join enc.service\_line t4

on t1.claim\_key=t4.claim\_key and t1.claim\_version\_key=t4.claim\_version\_key

where t3.hcc=t2.v23\_hcc and t1.claim\_key=t4.claim\_key and t1.claim\_version\_key=t4.claim\_version\_key;

**quit**;

**PROC** **SORT** DATA=hcc\_raps\_not\_enc\_er\_status\_fL1 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE TEST\_CPT AS

SELECT T1.\* ,CASE WHEN coalesce(T2.VALID\_FLAG,T3.VALID\_FLAG)='Y' THEN 'Y' ELSE '' END AS RAPS\_CPT\_FLAG,

coalesce(T2.DESCRIPTION,T3.DESCRIPTION) AS PROCEDURE\_DESC,

case when t1.place\_of\_service is not missing then 'PF'

WHEN t1.facility\_type ='21' then 'IP' ELSE 'OP' END AS CLAIM\_TYPE

FROM hcc\_raps\_not\_enc\_er\_status\_fL1 T1

LEFT JOIN DXHCCSTG.LKP\_CPT\_CODES t2 on T1.procedure\_code=T2.CPT

LEFT JOIN DXHCCSTG.LKP\_HCPCS\_CODES t3 on t1.procedure\_code=t3.hcpcs\_code

where T1.PROCEDURE\_CODE NOT IS MISSING;

**quit**;

**PROC** **SQL**;

CREATE TABLE VALID\_OPPF AS

SELECT DISTINCT t1.CLAIM\_VERSION\_KEY

FROM WORK.TEST\_CPT t1

WHERE t1.CPT\_FLAG = 'Y' AND t1.CLAIM\_TYPE IN

('OP','PF');

**QUIT**;

**PROC** **SQL**;

CREATE TABLE VALID\_IP AS

SELECT DISTINCT t1.CLAIM\_VERSION\_KEY

FROM WORK.TEST\_CPT t1

WHERE t1.CLAIM\_TYPE =('IP');

**QUIT**;

**DATA** VALID\_CVK;

SET VALID\_OPPF VALID\_IP;

**RUN**;

**PROC** **SQL**;

CREATE TABLE FLAG\_VALID\_CVK AS

SELECT t1.\*,

CASE WHEN t2.CLAIM\_VERSION\_KEY NOT IS MISSING THEN 'Y' ELSE '' END AS VALID\_CVK

FROM WORK.TEST\_CPT t1

LEFT JOIN WORK.VALID\_CVK t2 ON (t1.CLAIM\_VERSION\_KEY = t2.CLAIM\_VERSION\_KEY);

**QUIT**;

**DATA** hcc\_raps\_not\_enc\_er\_status\_fL2 OTHER;

SET FLAG\_VALID\_CVK;

IF VALID\_CVK='Y' THEN OUTPUT hcc\_raps\_not\_enc\_er\_status\_fL2;

ELSE OUTPUT OTHER;

**RUN**;

**PROC** **SORT** DATA=hcc\_raps\_not\_enc\_er\_status\_fL2 NODUPKEY; BY \_ALL\_;**RUN**;

/\* create a macro variable for cvk of interest\*/

**PROC** **SQL**;

create table dist\_cvk as

select distinct claim\_version\_key

from hcc\_raps\_not\_enc\_er\_status\_fL2;

**QUIT**;

**%Macro** \_nobs(dsn);

%Local dsid nobs rc opts;

%Let dsid = %SysFunc(Open(&dsn, I));

%If &dsid %Then %Do;

%Let nobs = %SysFunc(Attrn(&dsid,NLOBS));

%Let rc = %SysFunc(Close(&dsid));

%End; %Else %Do;

%Let nobs =.;

%End;

&nobs

**%Mend** \_nobs;

**%Macro** GetInfo(oramax=**1000**, dsn=, out=);

%Local i work\_dsn n\_ids;

%Let n\_ids = %\_Nobs(&dsn);

/\* Allow for more than max Oracle IN ids \*/

%Let work\_dsn=&out;

%Do i = **1** %To &n\_ids %By &oramax;

%Let last\_n = %Eval(&i + &oramax - 1);

Proc Sql Noprint;

Select put(claim\_version\_key,**12.0**) Into :ID\_LIST Separated by ','

/\*Select "'" ||PUT(strip(claim\_version\_key,$12.0)|| "'" Into :ID\_LIST Separated by ','\*/

From &dsn (FirstObs=&i obs=%SysFunc(MIN(&last\_n, &n\_ids)))

;

Quit;

%put &ID\_LIST;

proc sql ;

CREATE TABLE &work\_dsn AS

SELECT

clm.CLAIM\_VERSION\_KEY,

clm.RENDERING\_PROVIDER\_ID,

clm.WCN,

clm.claim\_key,

clm.seq\_claim\_id,

clm.claim\_type,

coalesce(CL2.PROVIDER\_TAXONOMY\_CODE, CL3.PROVIDER\_TAXONOMY\_CODE) as PROVIDER\_TAXONOMY\_CODE,

coalesce(CL2.FIRST\_NAME, CL3.FIRST\_NAME) as FIRST\_NAME,

coalesce(CL2.LAST\_NAME, CL3.LAST\_NAME) as LAST\_NAME,

coalesce(CL.ADDRESS\_LINE1, CL3.ADDRESS\_LINE1) as ADDRESS\_LINE1,

coalesce(CL.ADDRESS\_LINE2, CL3.ADDRESS\_LINE2) as ADDRESS\_LINE2,

coalesce(CL.CITY, CL3.CITY) as CITY,

coalesce(CL.STATE, CL3.STATE) as STATE,

coalesce(CL.ZIP\_CODE, CL3.ZIP\_CODE) as ZIP\_CODE,

T.COMMUNICATION\_NUMBER as PHONE\_NUMBER,

F.COMMUNICATION\_NUMBER as FAX\_NUMBER,

cpi.identifier as PROVIDER\_NPI

from ENC.CLAIM clm

left join ENC.CLAIM\_PROVIDER CL

on cl.CLAIM\_VERSION\_KEY = clm.CLAIM\_VERSION\_KEY

and cl.CLAIM\_KEY = clm.CLAIM\_KEY

and cl.CLAIM\_PROVIDER\_TYPE\_KEY = **9**

and CL.ADDRESS\_LINE1 is not null

left join ENC.CLAIM\_PROVIDER CL2

on cl2.CLAIM\_VERSION\_KEY = clm.CLAIM\_VERSION\_KEY

and cl2.CLAIM\_KEY = clm.CLAIM\_KEY

and clm.RENDERING\_PROVIDER\_ID=cl2.Provider\_sid

left join ENC.CLAIM\_PROVIDER CL3

on cl3.CLAIM\_VERSION\_KEY = clm.CLAIM\_VERSION\_KEY

and cl3.CLAIM\_KEY = clm.CLAIM\_KEY

and cl3.CLAIM\_PROVIDER\_TYPE\_KEY = **1**

left join ENC.CLAIM\_CONTACT T

on T.CLAIM\_VERSION\_KEY = clm.CLAIM\_VERSION\_KEY

and T.CLAIM\_KEY = clm.CLAIM\_KEY

and T.COMMUNICATION\_NUMBER\_QUALIFIER = 'TE'

and T.COMMUNICATION\_NUMBER is not null

left join ENC.CLAIM\_CONTACT F

on F.CLAIM\_VERSION\_KEY = clm.CLAIM\_VERSION\_KEY

and F.CLAIM\_KEY = clm.CLAIM\_KEY

and F.COMMUNICATION\_NUMBER\_QUALIFIER = 'FX'

and F.COMMUNICATION\_NUMBER is not null

left join ENC.CLAIM\_PROVIDER\_IDENTIFIER CPI

on CPI.Claim\_provider\_sid = clm.RENDERING\_PROVIDER\_ID

and cpi.Identifier\_qualifier = 'XX'

WHERE CLM.CLAIM\_VERSION\_KEY IN (&ID\_LIST)

;

quit;

%If &i Ne **1** %Then %Do;

Proc Append Base=&out Data=\_INFO; Run;

%End;

%Else %Do;

%Let work\_dsn = \_INFO;

%End;

%End;

Proc Sql Noprint;

Drop Table WORK.\_INFO;

Quit;

**%Mend** GetInfo;

%***GetInfo***(dsn=dist\_cvk, out=PROVIDER\_DETAIL);

;

**quit**;

**PROC** **SORT** DATA=PROVIDER\_DETAIL NODUPKEY;BY \_ALL\_;**RUN**;

**proc** **sql**;

create table hcc\_raps\_not\_enc\_er\_status\_fL3 as

select t1.\*,t2.reporting\_entity\_control\_number as ICN

from PROVIDER\_DETAIL t1

left join enc.claim\_outbound\_latest\_status t2

on t1.claim\_key=t2.claim\_key and t1.claim\_version\_key=t2.claim\_version\_key

where t1.claim\_key=t2.claim\_key and t1.claim\_version\_key=t2.claim\_version\_key ;

**quit**;

**proc** **sql**;

create table provider\_detail\_1 as

select t1.\*,t2.MBI,t2.LOB,t2.dos\_year,t2.HCC,t2.seq\_memb\_id,t2.esk

,t2.gender,t2.mbr\_last\_name,t2.mbr\_first\_name,t2.last\_modified\_date,

t2.dob,t2.mao004,t2.place\_of\_service,t2.diagnosis\_code,t2.procedure\_code

,t2.cpt\_flag,t2.procedure\_desc,t2.claim\_type,t2.valid\_cvk

from hcc\_raps\_not\_enc\_er\_status\_fL3 t1

left join hcc\_raps\_not\_enc\_er\_status\_fL2 t2

on t1.claim\_key=t2.claim\_key and t1.claim\_version\_key=t2.claim\_version\_key

;**quit**;

**PROC** **SORT** DATA=PROVIDER\_DETAIL\_1 NODUPKEY;BY \_ALL\_;**RUN**;

**proc** **sql**;

create table provider\_detail\_2 as

select t1.\*,t2.A\_D\_FLAG,T2.A\_D\_REASON\_CODE,

CASE WHEN T2.A\_D\_REASON\_CODE='H' THEN

'CPT/HCPCS code is not acceptable for risk adjustment'

WHEN T2.A\_D\_REASON\_CODE='N' THEN 'Facility not valid'

WHEN T2.A\_D\_REASON\_CODE='T' THEN 'Type of Bill is not acceptable for risk adjustment'

/\* when (t2.A\_D\_REASON\_CODE='' AND T2.CLAIM\_TYPE='IP') THEN 'Facility not valid'\*/

WHEN (T2.A\_D\_REASON\_COdE='' AND T2.claim\_type in ('OP','PF')) THEN 'CPT/HCPCS code is not acceptable for risk adjustment'

ELSE 'Facility not valid' END AS EDPS\_FILTERING\_LOGIC\_DESCRIPTION,

case when (t1.esk in (**225**,**2000**,**3000**) AND T2.ICN='') then 'Withheld from CMS submission - record issue'

when T1.esk in (**71**,**103**) then 'In Process'

when T1.esk=**703** then 'Submitted and Rejected by CMS'

when T1.esk=**602** then 'Submitted and Accepted by CMS'

when T1.esk=**604** then 'Waiting for CMS response'

when (t1.esk in (**225**,**3000**) and T2.ICN NOT IS MISSING) then 'Waiting for MAO-004'

ELSE '' END AS ENCOUNTER\_STATUS,

Case when t2.A\_D\_FLAG ='D' THEN 'Y' ELSE 'N' End AS MAO004\_RECEIVED,

'Y' as PASS\_RAPS\_FILTERING\_LOGIC,

'N' as PASS\_EDPS\_FILTERING\_LOGIC,

CATS(LAST\_NAME,',',FIRST\_NAME) AS PROVIDER\_NAME

from provider\_detail\_1 t1

left join REVREC.mao004\_phase3 t2

on t1.ICN=t2.ICN

;

**QUIT**;

**PROC** **SORT** DATA=PROVIDER\_DETAIL\_2 NODUPKEY;BY \_ALL\_;**RUN**;

**PROC** **SORT** DATA=PROVIDER\_DETAIL\_2 NODUPKEY;BY MBI HCC;**RUN**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Transpose not need for this code\*/

/\* TRANSPOSE THE PROCEDURE\_CODES\*/

**PROC** **SQL**;

CREATE VIEW INPUTDATA AS

SELECT DISTINCT claim\_version\_key, procedure\_code

FROM WORK.provider\_detail\_2

;

**QUIT**;

**PROC** **TRANSPOSE** DATA=INPUTDATA

OUT=OUTPUT\_TRANSPOSED

PREFIX=Column

NAME=Source

LABEL=Label

;

BY claim\_version\_key;

VAR procedure\_code ;

**RUN**; **QUIT**;

**PROC** **SQL**;

CREATE TABLE FINALTABLE AS

SELECT t1.claim\_version\_key, CATX(',',

t1.Column1,

t1.Column2,

t1.Column3,

t1.Column4,

t1.Column5,

t1.Column6,

t1.Column7,

t1.Column8,

t1.Column9,

t1.Column10,

t1.Column11,

t1.Column12,

t1.Column13,

t1.Column14,

t1.Column15,

t1.Column16,

t1.Column17,

t1.Column18,

t1.Column19,

t1.Column20,

t1.Column21,

t1.Column22,

t1.Column23,

t1.Column24,

t1.Column25,

t1.Column26,

t1.Column27,

t1.Column28,

t1.Column29,

t1.Column30,

t1.Column31,

t1.Column32,

t1.Column33,

t1.Column34,

t1.Column35,

t1.Column36,

t1.Column37,

t1.Column38,

t1.Column39,

t1.Column40,

t1.Column41,

t1.Column42,

t1.Column43,

t1.Column44,

t1.Column45,

t1.Column46,

t1.Column47,

t1.Column48,

t1.Column49,

t1.Column50,

t1.Column51,

t1.Column52,

t1.Column53,

t1.Column54,

t1.Column55

) AS procCONCAT

FROM WORK.OUTPUT\_TRANSPOSED t1;

**QUIT**;

**data** provider\_detail\_3(drop=procedure\_code);

set provider\_detail\_2;

**run**;

**proc** **sql**;

create table provider\_detail\_4 as

select distinct t1.\*,t2.procCONCAT as procedure\_codes

from provider\_detail\_3 t1 left join FINALTABLE t2

on t1.claim\_version\_key=t2.claim\_version\_key;

**quit**;

**data** HCC\_NOT\_IN\_ER\_RESEARCH\_REQUIRED;

set RAPSEDPS.HCC\_RAPS\_NOT\_ENC\_ER\_STATUS\_F ;

IF Processing\_Status='';

**run**;

/\*GET THE HICN FOR THE MBI\*/

**PROC** **SQL**;

CREATE TABLE RAPS\_UNIVERSE\_PY2019 AS

SELECT t1.HICN,

COALESCE(t3.MBI,t4.MBI,t1.HICN) AS MBI,

t1.lob,

t1.CLAIM\_ID,

t1.dx,

t1.from\_date,

t1.tru\_date,

t1.SOURCE,

t1.SUB\_SOURCE,

t1.type,

t1.DATA\_YEAR,

t2.HCC AS V23\_HCC,

CASE WHEN t1.TYPE IN ('01','02') THEN 'Y' ELSE 'N' END AS RAP\_IP

FROM REVREC.GJD\_RAPS\_RAW t1

INNER JOIN DXHCC.V23\_HCC\_NEW T2 ON T1.DX=T2.DX

LEFT JOIN REVREC.LKP\_MBI\_XWALK t3 ON (t1.HICN = t3.HICN)

LEFT JOIN REVREC.LKP\_MBI\_XWALK t4 ON (t1.HICN = t4.MBI)

WHERE t1.DATA\_YEAR = **2018** ;

**QUIT**;

**proc** **sql**;

create table match\_mbi as

select t1.\*,t2.lob,

t2.CLAIM\_ID,

t2.dx,

t2.from\_date,

t2.tru\_date,

t2.SOURCE,

t2.SUB\_SOURCE,

t2.type,

t2.DATA\_YEAR,

T2.V23\_HCC

from HCC\_NOT\_IN\_ER\_RESEARCH\_REQUIRED t1 left join RAPS\_UNIVERSE\_PY2019 t2

on t1.MBI=T2.MBI and t1.HCC=T2.V23\_HCC;

**QUIT**;

**PROC** **SORT** DATA=match\_mbi NODUPKEY; BY \_ALL\_;**RUN**;

**DATA** IT\_MBI\_20 IT\_MBI\_10 OTHER;

SET MATCH\_MBI;

IF (SOURCE='IT' AND TYPE=**20**) THEN OUTPUT IT\_MBI\_20;

ELSE IF (SOURCE='IT' AND TYPE=**10**) THEN OUTPUT IT\_MBI\_10;

ELSE OUTPUT OTHER;

**RUN**;

**DATA** IT\_MBI\_20\_1;

SET IT\_MBI\_20;

new = substr(CLAIM\_ID,**3** ,**9**)\***1**;

**RUN**;

**proc** **sql**;

create table test as

select distinct new

from IT\_MBI\_20\_1;

**QUIT**;

Libname instg oracle path=ODSPROD user="pmason" password="Encounters11)" schema=INSTG;**RUN**;

Libname WCEDI oracle path=ORCL user="Smotran" password="Options2019$" schema=WCEDI;**RUN**;

**proc** **sql**;

create table prof\_F\_status\_claims as

select T1.SEQ\_CLAIM\_ID,

T1.PRIMARY\_SVC\_DATE,

T1.CLAIM\_THRU\_DATE,

T1.LINE\_OF\_BUSINESS,

T1.DIAGNOSIS\_1,

T1.DIAGNOSIS\_2,

T1.DIAGNOSIS\_3,

T1.DIAGNOSIS\_4,

T1.DIAGNOSIS\_5,

T1.DIAGNOSIS\_6,

T1.DIAGNOSIS\_7,

T1.DIAGNOSIS\_8,

T1.DIAGNOSIS\_9,

T1.DIAGNOSIS\_10,

T1.DIAGNOSIS\_11,

T1.DIAGNOSIS\_12,

T2.SEQ\_PROV\_ID,

T2.SEQ\_MEMB\_ID,

T2.CLAIM\_STATUS,

T2.PROCESSING\_STATUS,

T2.CHECK\_DATE,

T2.POST\_DATE,

T3.MBI,

T3.HCC,

T3.CLAIM\_ID,

T3.DX,

T3.SOURCE,

T3.SUB\_SOURCE,

T3.TYPE,

T3.V23\_HCC

from instg.PROFSVC\_CLAIM\_HEADER\_MV T1

LEFT JOIN INSTG.PROFSVC\_CLAIM\_DETAIL\_MV T2 ON T1.SEQ\_CLAIM\_ID = T2.SEQ\_CLAIM\_ID

LEFT JOIN IT\_MBI\_20\_1 T3 ON T1.SEQ\_CLAIM\_ID = T3.NEW

WHERE T1.SEQ\_CLAIM\_ID = T3.NEW

AND T3.DX = T1.DIAGNOSIS\_1

OR T3.DX = T1.DIAGNOSIS\_2

OR T3.DX = T1.DIAGNOSIS\_3

OR T3.DX = T1.DIAGNOSIS\_4

OR T3.DX = T1.DIAGNOSIS\_5

OR T3.DX = T1.DIAGNOSIS\_6

OR T3.DX = T1.DIAGNOSIS\_7

OR T3.DX = T1.DIAGNOSIS\_8

OR T3.DX = T1.DIAGNOSIS\_9

OR T3.DX = T1.DIAGNOSIS\_10

OR T3.DX = T1.DIAGNOSIS\_11

OR T3.DX = T1.DIAGNOSIS\_12;

**run**;

**DATA** EDPSHCCCOMBINED;

SET RAPSEDPS.MAO004\_UNIVERSE\_HCC\_test\_2 RAPSEDPS.ER\_UNIVERSE\_HCC\_FINAL\_test\_2;

**RUN**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.EDPS\_HCC\_UNI\_test\_2 AS

SELECT DISTINCT t1.MBI,

t1.DOS\_YEAR,

t1.V23\_HCC AS HCC,

T2.AVG\_SCORE

FROM EDPSHCCCOMBINED t1

LEFT JOIN DXHCC.V22\_V23\_AVG\_SCORES T2 ON T1.V23\_HCC=T2.HCC;

**QUIT**;

**DATA** RAPSEDPS.RAPSEDPS\_HCC\_UNIVERSE\_test\_1;

SET RAPSEDPS.RAPS\_HCC\_UNI\_test\_2 RAPSEDPS.EDPS\_HCC\_UNI\_test\_2;

**RUN**;

**PROC** **SORT** DATA=RAPSEDPS.RAPSEDPS\_HCC\_UNIVERSE\_test\_1 NODUPKEY; BY \_ALL\_;**RUN**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_HCC\_test1 AS

SELECT DISTINCT t1.MBI,

t1.LOB,

t1.HPLAN,

t1.PY\_STATE,

t1.ACTIVE\_STATUS,

t1.PAYMENT\_MONTHS,

t2.DOS\_YEAR,

t2.HCC,

T2.AVG\_SCORE

FROM RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_test t1

JOIN RAPSEDPS.RAPSEDPS\_HCC\_UNIVERSE\_test\_1 t2 ON (t1.MBI = t2.MBI);

**QUIT**;

**PROC** **SQL**;

CREATE TABLE GET\_HCC\_LOW\_ALL AS

SELECT T1.\*,

T2.V22\_HCC\_LOW

FROM RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_HCC\_test1 T1

LEFT JOIN RAPSEDPS.V22\_HIERARCHIES T2 ON T1.HCC=T2.V22\_HCC\_HIGH WHERE T2.V22\_HCC\_LOW NOT IS MISSING;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE HCC\_UNI\_ALL AS

SELECT T1.\*,

t3.V22\_HCC\_High AS HCC\_HIGH, T4.V22\_HCC\_LOW AS HCC\_LOW

FROM RAPSEDPS.MEMBER\_UNIVERSE\_PY2019\_HCC\_test1 T1

LEFT JOIN GET\_HCC\_LOW\_ALL T2 ON T1.MBI=T2.MBI AND T1.HCC=T2.V22\_HCC\_LOW

LEFT JOIN RAPSEDPS.V22\_HIERARCHIES t3 ON t1.HCC=t3.V22\_HCC\_LOW

LEFT JOIN RAPSEDPS.V22\_HIERARCHIES t4 ON t1.HCC=t4.V22\_HCC\_HIGH

WHERE T2.V22\_HCC\_LOW is missing;

**QUIT**;

**PROC** **SQL**;

CREATE TABLE RAPSEDPS.MEMBER\_UNIV\_PY2019\_HCC\_FINAL1 AS

SELECT DISTINCT t1.MBI,

t1.LOB,

t1.HPLAN,

t1.PY\_STATE,

t1.ACTIVE\_STATUS,

t1.PAYMENT\_MONTHS,

t1.DOS\_YEAR,

t1.HCC,

t1.AVG\_SCORE AS HCC\_SCORE,

t1.HCC\_HIGH,

t2.AVG\_SCORE AS HCC\_SCORE\_HIGH,

t1.HCC\_LOW,

t3.AVG\_SCORE AS HCC\_SCORE\_LOW

FROM HCC\_UNI\_ALL t1

LEFT JOIN DXHCC.V22\_V23\_AVG\_SCORES t2 ON (t1.HCC\_HIGH = t2.HCC)

LEFT JOIN DXHCC.V22\_V23\_AVG\_SCORES t3 ON (t1.HCC\_LOW = t3.HCC);

**QUIT**;