

```

/*
    Set the following macro variables through %let or prompts.

    ldap_user
    ldap_pwd
    curr_month - e.g. 201509
*/

libname bmggeth teradata user="&ldap_user@LDAP" password="&ldap_pwd"
tdpid=bmg.wellsfargo.com schema=bmgu_bmggeth;
libname bmgpgeo teradata user="&ldap_user@LDAP" password="&ldap_pwd"
tdpid=bmg.wellsfargo.com schema= bmgpgeo;
libname claritas teradata user="&ldap_user@LDAP" password="&ldap_pwd"
tdpid=bmg.wellsfargo.com schema=bmgu_bmgclaritas;

/* Copy MDSS hierarchy table to SAS data set */
proc sql;
connect to teradata(server="BMG.WELLSFARGO.COM" user="&ldap_user.@LDAP"
password="&ldap_pwd" connection=global mode=teradata);
CREATE TABLE MDSS_hierarchy AS
select * from connection to teradata

    (select distinct AU
        , SUPER_GROUP_NM
        , GROUP_NM
        , REG_NM
        , TRTRY_NM
        , MKT_NM
/*        , store_type_desc */
        , ZIP
    from bmgu_bmggeth.view_stores_mdss
    where AU <> 0
        and status = 'O' /* Only include open or current AUs */
        and prim_store_to_outlet = 'Y'
        /*and store_type_desc = 'TRADITIONAL BRANCH'*/
        and curr_flag = 'Y'
    ) ;

disconnect from teradata;
quit;

/* Change AU from string to numeric */
data MDSS_hierarchy_final;
    set MDSS_hierarchy;
    AU_Num = input(AU,7.);
run;

proc sql noerrorstop;

/* Create primary checking data set */

```

```

/* Merge MDSS hierarchy with primary checking data */

create table primary_summary as
select
  b.asof_yyyymm,
  a.SUPER_GROUP_NM,
  a.GROUP_NM,
  a.REG_NM,
  a.TRTRY_NM,
  a.MKT_NM,
  b.eth_bmg_cd,
  (case
    when b.MILLENNIAL = 1 then 'Millennial'
    else 'Non-Millennial'
  end) as generation,
  sum(b.CNT) as fp_cnt,
  sum(case when b.primary_cust = 1 then cnt else 0 end) as prim_cnt
from MDSS_hierarchy_final a,
bmgeth.t_primary_cube b

where a.au_num=b.au_num
and a.SUPER_GROUP_NM <> 'OTHER'

group by 1,2,3,4,5,6,7,8
order by 1,2,3,4,5,6,7,8
;

/* Create new checking sales data set */

/* Merge MDSS hierarchy with primary checking data */

create table ncc_summary as
select
  b.asof_yyyymm,
  a.SUPER_GROUP_NM,
  a.GROUP_NM,
  a.REG_NM,
  a.TRTRY_NM,
  a.MKT_NM,
  b.eth_bmg_cd,
  (case
    when b.MILLENNIAL = 1 then 'Millennial'
    else 'Non-Millennial'
  end) as generation,
  /* a.zip, */
  sum(b.CNT) as all_sales_cnt,
  sum(case when b.NEW_PROD_CUST_IND = 1 then cnt else 0 end) as ncc_cnt
from MDSS_hierarchy_final a,
bmgeth.t_ncc_cube b

```

```

where a.au_num=b.au_num
and a.SUPER_GROUP_NM <> 'OTHER'

group by 1,2,3,4,5,6,7,8
order by 1,2,3,4,5,6,7,8
;

/* Combine into one summary table */
create table dda_summary as
select
  a.asof_yyyymm,
  a.SUPER_GROUP_NM,
  a.GROUP_NM,
  a.REG_NM,
  a.TRTRY_NM,
  a.MKT_NM,
  a.eth_bmg_cd,
  a.generation,
  a.fp_cnt,
  a.prim_cnt,
  b.all_sales_cnt,
  b.ncc_cnt
from primary_summary a

left join ncc_summary b
on
a.asof_yyyymm = b.asof_yyyymm and
a.SUPER_GROUP_NM = b.SUPER_GROUP_NM and
a.GROUP_NM = b.GROUP_NM and
a.REG_NM = b.REG_NM and
a.TRTRY_NM = b.TRTRY_NM and
a.MKT_NM = b.MKT_NM and
a.eth_bmg_cd = b.eth_bmg_cd and
a.generation = b.generation
;

/* When zips appear in multiple markets, weight population */

create table mkt_zip_map as
select
  a.SUPER_GROUP_NM,
  a.GROUP_NM,
  a.REG_NM,
  a.TRTRY_NM,
  a.MKT_NM,
  a.zip
from MDSS_hierarchy_final a,
bmgeth.t_primary_cube b

where a.au_num=b.au_num
and a.SUPER_GROUP_NM <> 'OTHER'
and asof_yyyymm = &curr_month /* for regions, use current month in cube

```

```

matching */

group by 1,2,3,4,5,6
;

/* Current Year demographics */

create table pop_summary as
select
SUPER_GROUP_NM,
GROUP_NM,
REG_NM,
TRTRY_NM,
MKT_NM,
sum(x.pop*y.num/y.denom) as pop,
sum(x.pop_18p*y.num/y.denom) as pop_18p,
sum(x.black*y.num/y.denom) as black,
sum(x.blk_18p*y.num/y.denom) as blk_pop_18p,
sum(x.hispanic*y.num/y.denom) as hispanic,
sum(x.hsp_18p*y.num/y.denom) as hsp_pop_18p,
sum(x.asian*y.num/y.denom) as asian,
sum(x.asi_18p*y.num/y.denom) as asi_pop_18p,
sum(x.asi_asind*y.num/y.denom) as asi_asind,
sum(x.asi_chine*y.num/y.denom) as asi_chine,
sum(x.asi_filip*y.num/y.denom) as asi_filip,
sum(x.asi_korea*y.num/y.denom) as asi_korea,
sum(x.asi_japan*y.num/y.denom) as asi_japan,
sum(x.asi_viet*y.num/y.denom) as asi_viet,
sum((x.pop_18_20+x.pop_21_24+x.pop_25_34)*y.num/y.denom) as pop_mill,
sum((x.blk_18_20+x.blk_21_24+x.blk_25_34)*y.num/y.denom) as
blk_pop_mill,
sum((x.asi_18_20+x.asi_21_24+x.asi_25_34)*y.num/y.denom) as
asi_pop_mill,
sum((x.hsp_18_20+x.hsp_21_24+x.hsp_25_34)*y.num/y.denom) as
hsp_pop_mill
from claritas.t_zip_popfact_2016CY x,
(select
a.SUPER_GROUP_NM,
a.GROUP_NM,
a.REG_NM,
a.TRTRY_NM,
a.MKT_NM,
a.zip,
a.num,
b.denom
from
(select
SUPER_GROUP_NM,
GROUP_NM,
REG_NM,
TRTRY_NM,
MKT_NM,

```

```

zip,
count(*) as num
from mkt_zip_map

group by 1,2,3,4,5,6) a,
(select
zip,
count(*) as denom
from mkt_zip_map

group by 1) b

where a.zip=b.zip) y

where x.zip=y.zip

group by 1,2,3,4,5
order by 1,2,3,4,5
;

/* 5 Year projection demographics */

create table pop_summary_5y as
select
SUPER_GROUP_NM,
GROUP_NM,
REG_NM,
TRTRY_NM,
MKT_NM,
sum(x.pop_5y*y.num/y.denom) as pop_5y,
sum(x.pop_18p_5y*y.num/y.denom) as pop_18p_5y,
sum(x.black_5y*y.num/y.denom) as black_5y,
sum(x.blk_18p_5y*y.num/y.denom) as blk_pop_18p_5y,
sum(x.hispanic_5y*y.num/y.denom) as hispanic_5y,
sum(x.hsp_18p_5y*y.num/y.denom) as hsp_pop_18p_5y,
sum(x.asian_5y*y.num/y.denom) as asian_5y,
sum(x.asi_18p_5y*y.num/y.denom) as asi_18p_5y,
sum(x.asi_asind_5y*y.num/y.denom) as asi_asind_5y,
sum(x.asi_chine_5y*y.num/y.denom) as asi_chine_5y,
sum(x.asi_filip_5y*y.num/y.denom) as asi_filip_5y,
sum(x.asi_korea_5y*y.num/y.denom) as asi_korea_5y,
sum(x.asi_japan_5y*y.num/y.denom) as asi_japan_5y,
sum(x.asi_viet_5y*y.num/y.denom) as asi_viet_5y,
sum((x.pop_18_20_5y+x.pop_21_24_5y+x.pop_25_34_5y)*y.num/y.denom) as
pop_mill_5y,
sum((x.blk_18_20_5y+x.blk_21_24_5y+x.blk_25_34_5y)*y.num/y.denom) as
blk_pop_mill_5y,
sum((x.asi_18_20_5y+x.asi_21_24_5y+x.asi_25_34_5y)*y.num/y.denom) as
asi_pop_mill,
sum((x.hsp_18_20_5y+x.hsp_21_24_5y+x.hsp_25_34_5y)*y.num/y.denom) as
hsp_pop_mill
from claritas.t_zip_popfact_20165Y x,

```

```

(select
a.SUPER_GROUP_NM,
a.GROUP_NM,
a.REG_NM,
a.TRTRY_NM,
a.MKT_NM,
a.zip,
a.num,
b.denom
from
(select
SUPER_GROUP_NM,
GROUP_NM,
REG_NM,
TRTRY_NM,
MKT_NM,
zip,
count(*) as num
from mkt_zip_map

group by 1,2,3,4,5,6) a,
(select
zip,
count(*) as denom
from mkt_zip_map

group by 1) b

where a.zip=b.zip) y

where x.zip=y.zip

group by 1,2,3,4,5
order by 1,2,3,4,5
;

quit;

/*
Step 2

Build Asian DDA sub-segment spreadsheet for Shuyi.
*/

/* Merge WF counts with Nielsen pop facts */
data final;
merge pop_summary (in=in1)
      pop_summary_5Y (in=in2)
      dda_summary (in=in3 where=(asof_yyyymm=&curr_month));
by SUPER_GROUP_NM GROUP_NM REG_NM TRTRY_NM MKT_NM;
if in1 and in2 and in3;
run;

```

```

/* Create dataset with columns in the order of the final spreadsheet */
proc sql noerrorstop;

create table asian_dda_final as
select
SUPER_GROUP_NM label='Column A - Organization',
GROUP_NM label='Column B - Lead Region',
REG_NM label='Column C - Region',
TRTRY_NM label='Column D - Territory',
MKT_NM label='Column E - Market',
/* Claritas population, no adult population for asian sub-segments */
max(pop_18p) as pop_18p label='Column F - Adult Pop',
max(pop) as pop label='Column G - Pop',
max(asi_pop_18p) as asi_pop_18p label='Column H - Asian Adult Pop',
max(asian) as asian label='Column I - Asian Pop',
max(asi_asind) as asi_asind label='Column J - Asian Indian Pop',
max(asi_chine) as asi_chine label='Column K - Chinese Pop',
max(asi_filip) as asi_filip label='Column L - Filipino Pop',
max(asi_korea) as asi_korea label='Column M - Korean Pop',
max(asi_japan) as asi_japan label='Column N - Japanese Pop',
max(asi_viet) as asi_viet label='Column O - Vietnamese Pop',

/* Overall Counts that go in all Asian sections */
sum(fp_cnt) as fp_cnt label='Columns Z, AF, AL, AR, AX, BD, BJ, BP - Total
Checking',
sum(ncc_cnt) as ncc_cnt label='Columns AC, AI, AO, AU, BA, BG, BM, BS -
NCC',

/* Asian sub-segment sections */

/* All Asian */
sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then fp_cnt else 0 end)
as fp_cnt_all_asian label='Column Y - Total Checking All Asian',
sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then ncc_cnt else 0 end)
as ncc_cnt_all_asian label='Column AB - NCC All Asian',
/* Asian Indian */
sum(case when ETH_BMG_CD = 8 then fp_cnt else 0 end) as fp_cnt_AI
label='Column AE - Total Checking Asian Indian',
sum(case when ETH_BMG_CD = 8 then ncc_cnt else 0 end) as ncc_cnt_AI
label='Column AH - NCC Asian Indian',
/* Chinese */
sum(case when ETH_BMG_CD = 9 then fp_cnt else 0 end) as fp_cnt_Chinese
label='Column AK - Total Checking Chinese',
sum(case when ETH_BMG_CD = 9 then ncc_cnt else 0 end) as ncc_cnt_Chinese
label='Column AN - NCC Chinese',
/* Filipino */
sum(case when ETH_BMG_CD = 10 then fp_cnt else 0 end) as fp_cnt_Filipino
label='Column AQ - Total Checking Filipino',
sum(case when ETH_BMG_CD = 10 then ncc_cnt else 0 end) as ncc_cnt_Filipino
label='Column AT - NCC Filipino',
/* Korean */

```

```

sum(case when ETH_BMG_CD = 11 then fp_cnt else 0 end) as fp_cnt_Korean
label='Column AW - Total Checking Korean',
sum(case when ETH_BMG_CD = 11 then ncc_cnt else 0 end) as ncc_cnt_Korean
label='Column AZ - NCC Korean',
/* Japanese */
sum(case when ETH_BMG_CD = 13 then fp_cnt else 0 end) as fp_cnt_Japan
label='Column BC - Total Checking Japanese',
sum(case when ETH_BMG_CD = 13 then ncc_cnt else 0 end) as ncc_cnt_Japan
label='Column BF - NCC Japanese',
/* Vietnamese */
sum(case when ETH_BMG_CD = 12 then fp_cnt else 0 end) as fp_cnt_Viet
label='Column BI - Total Checking Vietnamese',
sum(case when ETH_BMG_CD = 12 then ncc_cnt else 0 end) as ncc_cnt_Viet
label='Column BL - NCC Vietnamese',
/* Other Asian */
sum(case when ETH_BMG_CD in (2,14) then fp_cnt else 0 end) as
fp_cnt_other_asian label='Column BO - Total Checking All Other Asian',
sum(case when ETH_BMG_CD in (2,14) then ncc_cnt else 0 end) as
ncc_cnt_other_asian label='Column BR - NCC All Other Asian',

/* Columns for Primary section */
sum(prim_cnt) as prim_cnt label='Column BU - Primary Checking',
sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then prim_cnt else 0
end) as prim_cnt_all_asian label='Column BV - Primary Checking',
sum(case when ETH_BMG_CD = 8 then prim_cnt else 0 end) as prim_cnt_AI
label='Column BW - Primary Checking Asian Indian',
sum(case when ETH_BMG_CD = 9 then prim_cnt else 0 end) as prim_cnt_Chinese
label='Column BX - Primary Checking Chinese',
sum(case when ETH_BMG_CD = 10 then prim_cnt else 0 end) as prim_cnt_Filipino
label='Column BY - Primary Checking Filipino',
sum(case when ETH_BMG_CD = 11 then prim_cnt else 0 end) as prim_cnt_Korean
label='Column BZ - Primary Checking Korean',
sum(case when ETH_BMG_CD = 13 then prim_cnt else 0 end) as prim_cnt_Japan
label='Column CA - Primary Checking Japanese',
sum(case when ETH_BMG_CD = 12 then prim_cnt else 0 end) as prim_cnt_Viet
label='Column CB - Primary Checking Vietnamese',
sum(case when ETH_BMG_CD in (2,14) then prim_cnt else 0 end) as
prim_cnt_other_asian label='Column CC - Primary Checking All Other Asian',

/* Columns for All Checking Sales section */

sum(all_sales_cnt) as all_sales_cnt label='Column CD - Checking Sales',
sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then all_sales_cnt else
0 end) as all_sales_cnt_all_asian label='Column CE - Checking Sales All
Asian',
sum(case when ETH_BMG_CD = 8 then all_sales_cnt else 0 end) as
all_sales_cnt_AI label='Column CF - Checking Sales Asian Indian',
sum(case when ETH_BMG_CD = 9 then all_sales_cnt else 0 end) as
all_sales_cnt_Chinese label='Column CG - Checking Sales Chinese',
sum(case when ETH_BMG_CD = 10 then all_sales_cnt else 0 end) as
all_sales_cnt_Filipino label='Column CH - Checking Sales Filipino',
sum(case when ETH_BMG_CD = 11 then all_sales_cnt else 0 end) as

```



```

all_sales_cnt_Korean label='Column CI - Checking Sales Korean',
sum(case when ETH_BMG_CD = 13 then all_sales_cnt else 0 end) as
all_sales_cnt_Japan label='Column CJ - Checking Sales Japanese',
sum(case when ETH_BMG_CD = 12 then all_sales_cnt else 0 end) as
all_sales_cnt_Viet label='Column CK - Checking Sales Vietnamese',
sum(case when ETH_BMG_CD in (2,14) then all_sales_cnt else 0 end) as
all_sales_cnt_other_asian label='Column CL - Checking Sales All Other
Asian'

from final

group by 1,2,3,4,5
order by 1,2,3,4,5
;

quit;

/* Make less cluttered--remove all "noise" markets like business banking
centers, etc, */
data asian_dda_final; set asian_dda_final(where=(all_sales_cnt is not
null)); run;

/* Write out spreadsheet with counts. Use labels as column headings */
proc export
    data=asian_dda_final
    dlabel
    dbms=xlsx

outfile='/sas/AU48750/BusinessAnalytics/QuarterlyDDA/data/asian_dda_fi
nal.xlsx'
    replace;
run;

/*
    Step 3 - Build Asian sub-segment by DMA
*/

proc sql noerrorstop;

/* Create primary checking data set */

/* Merge MDSS hierarchy with primary checking data */

create table primary_summary_dma as
select
    b.zip,
    b.eth_bmg_cd,
    (case
        when b.MILLENNIAL = 1 then 'Millennial'
        else 'Non-Millennial'
    )

```

```

        end) as generation,
        sum(b.CNT) as fp_cnt,
        sum(case when b.primary_cust = 1 then cnt else 0 end) as prim_cnt
from MDSS_hierarchy_final a,
bmgeth.t_primary_cube b

where a.au_num=b.au_num
and a.SUPER_GROUP_NM <> 'OTHER' /* remove OTHER region */
and b.asof_yyyymm = &curr_month

group by 1,2,3
order by 1,2,3
;

create table primary_zip as
select
distinct zip
from primary_summary_dma
;

/* Create new checking sales data set */

/* Merge MDSS hierarchy with primary checking data */

create table ncc_summary_dma as
select
    b.zip,
    b.eth_bmg_cd,
    (case
        when b.MILLENNIAL = 1 then 'Millennial'
        else 'Non-Millennial'
    end) as generation,
/* a.zip, */
    sum(b.CNT) as all_sales_cnt,
    sum(case when b.NEW_PROD_CUST_IND = 1 then cnt else 0 end) as ncc_cnt
from MDSS_hierarchy_final a,
bmgeth.t_ncc_cube b

where a.au_num=b.au_num
and a.SUPER_GROUP_NM <> 'OTHER' /* remove OTHER region */
and b.asof_yyyymm = &curr_month

group by 1,2,3
order by 1,2,3
;

/* Combine into one summary table */
create table dda_summary_dma as
select
    c.dma_name,
    a.eth_bmg_cd,

```

```

a.generation,
sum(a.fp_cnt) as fp_cnt,
sum(a.prim_cnt) as prim_cnt,
sum(b.all_sales_cnt) as all_sales_cnt,
sum(b.ncc_cnt) as ncc_cnt
from primary_summary_dma a

```

```

left join ncc_summary_dma b
on
a.zip = b.zip and
a.eth_bmg_cd = b.eth_bmg_cd and
a.generation = b.generation

```

```

left join (
select
zip,
dma_name
from bmgpgeo.ref_zip5_hist_bmg
where asof_yyyymm = &curr_month) c
on a.zip=c.zip

```

```

group by 1,2,3
order by 1,2,3
;

```

```

create table pop_summary_dma as
select
z.dma_name,
sum(x.pop) as pop,
sum(x.pop_18p) as pop_18p,
sum(x.black) as black,
sum(x.blk_18p) as blk_pop_18p,
sum(x.hispanic) as hispanic,
sum(x.hsp_18p) as hsp_pop_18p,
sum(x.asian) as asian,
sum(x.asi_18p) as asi_pop_18p,
sum(x.asi_asind) as asi_asind,
sum(x.asi_chine) as asi_chine,
sum(x.asi_filip) as asi_filip,
sum(x.asi_korea) as asi_korea,
sum(x.asi_japan) as asi_japan,
sum(x.asi_viet) as asi_viet,
sum(x.pop_18_20+x.pop_21_24+x.pop_25_34) as pop_mill,
sum(x.blk_18_20+x.blk_21_24+x.blk_25_34) as blk_pop_mill,
sum(x.asi_18_20+x.asi_21_24+x.asi_25_34) as asi_pop_mill,
sum(x.hsp_18_20+x.hsp_21_24+x.hsp_25_34) as hsp_pop_mill
from claritas.t_zip_popfact_2016CY x

```

```

inner join (
select
distinct zip

```

```
from primary_summary_dma) y
on input(x.zip,7.)=y.zip
```

```
left join (
select
dma_cd,
dma_name,
count(*)
from bmgpgeo.ref_zip5_hist_bmg
where asof_yyyymm = &curr_month
and dma_cd is not null
group by 1,2) z
on input(x.dma_code,3.)=z.dma_cd
```

```
group by 1
order by 1
;
```

```
quit;
```

```
/* Merge WF counts with Nielsen pop facts */
```

```
data final_dma;
    merge pop_summary_dma (in=in1)
          dda_summary_dma (in=in2);
    by dma_name;
    if in1 and in2;
run;
```

```
proc sql noerrorstop;
```

```
create table asian_dda_final_dma as
select
dma_name label='Column A - DMA NAME',
/* Claritas population, no adult population for asian sub-segments */
max(pop_18p) as pop_18p label='Column F - Adult Pop',
max(pop) as pop label='Column G - Pop',
max(asi_pop_18p) as asi_pop_18p label='Column H - Asian Adult Pop',
max(asian) as asian label='Column I - Asian Pop',
max(asi_asind) as asi_asind label='Column J - Asian Indian Pop',
max(asi_chine) as asi_chine label='Column K - Chinese Pop',
max(asi_filip) as asi_filip label='Column L - Filipino Pop',
max(asi_korea) as asi_korea label='Column M - Korean Pop',
max(asi_japan) as asi_japan label='Column N - Japanese Pop',
max(asi_viet) as asi_viet label='Column O - Vietnamese Pop',

/* Overall Counts that go in all Asian sections */
sum(fp_cnt) as fp_cnt label='Columns Z, AF, AL, AR, AX, BD, BJ, BP - Total
Checking',
sum(ncc_cnt) as ncc_cnt label='Columns AC, AI, AO, AU, BA, BG, BM, BS -
```

```

NCC',

/* Asian sub-segment sections */

/* All Asian */
sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then fp_cnt else 0 end)
as fp_cnt_all_asian label='Column Y - Total Checking All Asian',
sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then ncc_cnt else 0 end)
as ncc_cnt_all_asian label='Column AB - NCC All Asian',
/* Asian Indian */
sum(case when ETH_BMG_CD = 8 then fp_cnt else 0 end) as fp_cnt_AI
label='Column AE - Total Checking Asian Indian',
sum(case when ETH_BMG_CD = 8 then ncc_cnt else 0 end) as ncc_cnt_AI
label='Column AH - NCC Asian Indian',
/* Chinese */
sum(case when ETH_BMG_CD = 9 then fp_cnt else 0 end) as fp_cnt_Chinese
label='Column AK - Total Checking Chinese',
sum(case when ETH_BMG_CD = 9 then ncc_cnt else 0 end) as ncc_cnt_Chinese
label='Column AN - NCC Chinese',
/* Filipino */
sum(case when ETH_BMG_CD = 10 then fp_cnt else 0 end) as fp_cnt_Filipino
label='Column AQ - Total Checking Filipino',
sum(case when ETH_BMG_CD = 10 then ncc_cnt else 0 end) as ncc_cnt_Filipino
label='Column AT - NCC Filipino',
/* Korean */
sum(case when ETH_BMG_CD = 11 then fp_cnt else 0 end) as fp_cnt_Korean
label='Column AW - Total Checking Korean',
sum(case when ETH_BMG_CD = 11 then ncc_cnt else 0 end) as ncc_cnt_Korean
label='Column AZ - NCC Korean',
/* Japanese */
sum(case when ETH_BMG_CD = 13 then fp_cnt else 0 end) as fp_cnt_Japan
label='Column BC - Total Checking Japanese',
sum(case when ETH_BMG_CD = 13 then ncc_cnt else 0 end) as ncc_cnt_Japan
label='Column BF - NCC Japanese',
/* Vietnamese */
sum(case when ETH_BMG_CD = 12 then fp_cnt else 0 end) as fp_cnt_Viet
label='Column BI - Total Checking Vietnamese',
sum(case when ETH_BMG_CD = 12 then ncc_cnt else 0 end) as ncc_cnt_Viet
label='Column BL - NCC Vietnamese',
/* Other Asian */
sum(case when ETH_BMG_CD in (2,14) then fp_cnt else 0 end) as
fp_cnt_other_asian label='Column BO - Total Checking All Other Asian',
sum(case when ETH_BMG_CD in (2,14) then ncc_cnt else 0 end) as
ncc_cnt_other_asian label='Column BR - NCC All Other Asian',

/* Columns for Primary section */
sum(prim_cnt) as prim_cnt label='Column BU - Primary Checking',
sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then prim_cnt else 0
end) as prim_cnt_all_asian label='Column BV - Primary Checking',
sum(case when ETH_BMG_CD = 8 then prim_cnt else 0 end) as prim_cnt_AI
label='Column BW - Primary Checking Asian Indian',
sum(case when ETH_BMG_CD = 9 then prim_cnt else 0 end) as prim_cnt_Chinese

```

```

label='Column BX - Primary Checking Chinese',
sum(case when ETH_BMG_CD = 10 then prim_cnt else 0 end) as prim_cnt_Filipino
label='Column BY - Primary Checking Filipino',
sum(case when ETH_BMG_CD = 11 then prim_cnt else 0 end) as prim_cnt_Korean
label='Column BZ - Primary Checking Korean',
sum(case when ETH_BMG_CD = 13 then prim_cnt else 0 end) as prim_cnt_Japan
label='Column CA - Primary Checking Japanese',
sum(case when ETH_BMG_CD = 12 then prim_cnt else 0 end) as prim_cnt_Viet
label='Column CB - Primary Checking Vietnamese',
sum(case when ETH_BMG_CD in (2,14) then prim_cnt else 0 end) as
prim_cnt_other_asian label='Column CC - Primary Checking All Other Asian',

/* Columns for All Checking Sales section */

sum(all_sales_cnt) as all_sales_cnt label='Column CD - Checking Sales',
sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then all_sales_cnt else
0 end) as all_sales_cnt_all_asian label='Column CE - Checking Sales All
Asian',
sum(case when ETH_BMG_CD = 8 then all_sales_cnt else 0 end) as
all_sales_cnt_AI label='Column CF - Checking Sales Asian Indian',
sum(case when ETH_BMG_CD = 9 then all_sales_cnt else 0 end) as
all_sales_cnt_Chinese label='Column CG - Checking Sales Chinese',
sum(case when ETH_BMG_CD = 10 then all_sales_cnt else 0 end) as
all_sales_cnt_Filipino label='Column CH - Checking Sales Filipino',
sum(case when ETH_BMG_CD = 11 then all_sales_cnt else 0 end) as
all_sales_cnt_Korean label='Column CI - Checking Sales Korean',
sum(case when ETH_BMG_CD = 13 then all_sales_cnt else 0 end) as
all_sales_cnt_Japan label='Column CJ - Checking Sales Japanese',
sum(case when ETH_BMG_CD = 12 then all_sales_cnt else 0 end) as
all_sales_cnt_Viet label='Column CK - Checking Sales Vietnamese',
sum(case when ETH_BMG_CD in (2,14) then all_sales_cnt else 0 end) as
all_sales_cnt_other_asian label='Column CL - Checking Sales All Other
Asian'

from final_dma

group by 1
order by 1
;

quit;

/* Make less cluttered--remove all "noise" markets like business banking
centers, etc, */
data asian_dda_final_dma; set asian_dda_final_dma(where=(all_sales_cnt
is not null)); run;

/* Write out spreadsheet with counts. Use labels as column headings */
proc export
    data=asian_dda_final_dma
    dlabel
    dbms=xlsx

```

```

outfile='/sas/AU48750/BusinessAnalytics/QuarterlyDDA/data/asian_dda_final_dma.xlsx'
    replace;
run;

quit;

/*
    Step 4 - Build Millennial DDA for Alec Hughes
*/
proc sql noerrorstop;

create table mill_dda_final as
select
SUPER_GROUP_NM label='Column A - Organization',
GROUP_NM label='Column B - Lead Region',
REG_NM label='Column C - Region',
TRTRY_NM label='Column D - Territory',
MKT_NM label='Column E - Market',
/* Claritas population, no adult population for asian sub-segments */
max(pop_18p) as pop_18p label='Column F - Adult Pop',
max(pop_mill) as pop_mill label='Column G - Millennial Pop',
sum(case when generation = 'Millennial' then fp_cnt else 0 end) as
fp_cnt_mill label='Column I - Total Checking Millennial',
sum(fp_cnt) as fp_cnt label='Columns J - Total Checking',
sum(case when generation = 'Millennial' then ncc_cnt else 0 end) as
ncc_cnt_mill label='Column L - NCC Millennial',
sum(ncc_cnt) as ncc_cnt label='Column M - NCC',
sum(prim_cnt) as prim_cnt label='Column O - Primary Checking',
sum(case when generation = 'Millennial' then prim_cnt else 0 end) as
prim_cnt_mill label='Column P - Primary Checking Millennial',
sum(all_sales_cnt) as all_sales_cnt label='Column Q - Checking Sales',
sum(case when generation = 'Millennial' then all_sales_cnt else 0 end) as
all_sales_cnt_mill label='Column R - All Sales Checking Millennial'
from final

group by 1,2,3,4,5
order by 1,2,3,4,5
;

quit;

/* Make less cluttered--remove all "noise" markets like business banking
centers, etc, */
data mill_dda_final; set mill_dda_final(where=(all_sales_cnt is not
null)); run;

/* Write out spreadsheet with counts. Use labels as column headings */
proc export

```

```

        data=mill_dda_final
        dblabel
        dbms=xlsx

outfile='/sas/AU48750/BusinessAnalytics/QuarterlyDDA/data/mill_dda_fin
al.xlsx'
        replace;
run;

/*
    The following data goes into the "Metric Final" data tab of the Checking
    Customer Dashboard
*/

proc sql noerrorstop;

/*
    Create the Primary Checking Dashboard
*/

create table primary_temp as
select
    asof_yyyymm,
    SUPER_GROUP_NM as Organization,
    GROUP_NM as LeadRegion,
    REG_NM as Region,
    TRTRY_NM as Territory,
    MKT_NM as Market,
    Generation,
    sum(fp_cnt) as cnt,
    sum(prim_cnt) as prim_cnt,
    sum(case when ETH_BMG_CD = 1 then fp_cnt else 0 end) as AA_cnt,
    sum(case when ETH_BMG_CD = 1 then prim_cnt else 0 end) as AA_prim_cnt,
    sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then fp_cnt else 0
end) as Asian_cnt,
    sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then prim_cnt else
0 end) as Asian_prim_cnt,
    sum(case when ETH_BMG_CD = 4 then fp_cnt else 0 end) as Hisp_cnt,
    sum(case when ETH_BMG_CD = 4 then prim_cnt else 0 end) as Hisp_prim_cnt
from primary_summary

group by 1,2,3,4,5,6,7
order by 1,2,3,4,5,6,7
;

create table primary_dashboard as
select
    a.*,
    /* Divide adult population into Millennial and Non-Millennial counts */
    (case
        when a.Generation = 'Millennial' then b.pop_mill
        else (b.pop_18p-b.pop_mill)

```



```

    end) as pop,
  (case
    when a.Generation = 'Millennial' then b.blk_pop_mill
    else (b.blk_pop_18p-b.blk_pop_mill)
  end) as AA_pop,
  (case
    when a.Generation = 'Millennial' then b.asi_pop_mill
    else (b.asi_pop_18p-b.asi_pop_mill)
  end) as Asian_pop,
  (case
    when a.Generation = 'Millennial' then b.hsp_pop_mill
    else (b.hsp_pop_18p-b.hsp_pop_mill)
  end) as Hisp_pop
from primary_temp a,
pop_summary b

where a.Organization = b.SUPER_GROUP_NM
and a.LeadRegion = b.GROUP_NM
and a.Region = b.REG_NM
and a.Territory = b.TRTRY_NM
and a.Market = b.MKT_NM
;

/*
  Create the New Checking Dashboard
*/

create table ncc_temp as
select
  asof_yyyymm,
  SUPER_GROUP_NM as Organization,
  GROUP_NM as LeadRegion,
  REG_NM as Region,
  TRTRY_NM as Territory,
  MKT_NM as Market,
  Generation,
  sum(all_sales_cnt) as cnt,
  sum(ncc_cnt) as ncc_cnt,
  sum(case when ETH_BMG_CD = 1 then all_sales_cnt else 0 end) as AA_cnt,
  sum(case when ETH_BMG_CD = 1 then ncc_cnt else 0 end) as AA_ncc_cnt,
  sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then all_sales_cnt
else 0 end) as Asian_cnt,
  sum(case when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then ncc_cnt else 0
end) as Asian_ncc_cnt,
  sum(case when ETH_BMG_CD = 4 then all_sales_cnt else 0 end) as Hisp_cnt,
  sum(case when ETH_BMG_CD = 4 then ncc_cnt else 0 end) as Hisp_ncc_cnt
from ncc_summary

group by 1,2,3,4,5,6,7
order by 1,2,3,4,5,6,7
;

```

```

/* Merge pop facts with new checking sales data */

create table ncc_dashboard as
select
  a.*,
  /* Divide adult population into Millennial and Non-Millennial counts */
  (case
    when a.generation = 'Millennial' then b.pop_mill
    else (b.pop_18p-b.pop_mill)
  end) as pop,
  (case
    when a.generation = 'Millennial' then b.blk_pop_mill
    else (b.blk_pop_18p-b.blk_pop_mill)
  end) as AA_pop,
  (case
    when a.generation = 'Millennial' then b.asi_pop_mill
    else (b.asi_pop_18p-b.asi_pop_mill)
  end) as Asian_pop,
  (case
    when a.generation = 'Millennial' then b.hsp_pop_mill
    else (b.hsp_pop_18p-b.hsp_pop_mill)
  end) as Hisp_pop
from ncc_temp a,
pop_summary b

where a.Organization = b.SUPER_GROUP_NM
and a.LeadRegion = b.GROUP_NM
and a.Region = b.REG_NM
and a.Territory = b.TRTRY_NM
and a.Market = b.MKT_NM
;

/*
  The following
*/

create table metric_final as

select
  a.asof_yyyymm,
  a.SUPER_GROUP_NM as Organization,
  a.GROUP_NM as LeadRegion,
  a.REG_NM as Region,
  a.TRTRY_NM as Territory,
  a.MKT_NM as Market,
  'First Position Checking Customer' as Metric,
  (case when ETH_BMG_CD = 1 then 'African American'
    when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then 'Asian'
    when ETH_BMG_CD = 4 then 'Hispanic'
    else 'Other'
  )

```

```

        end) as ethnicity,
        generation,
        sum(fp_cnt) as cnt
from primary_summary a,
pop_summary b

where a.SUPER_GROUP_NM = b.SUPER_GROUP_NM
and a.GROUP_NM = b.GROUP_NM
and a.REG_NM = b.REG_NM
and a.TRTRY_NM = b.TRTRY_NM
and a.MKT_NM = b.MKT_NM

group by 1,2,3,4,5,6,7,8,9

union all

select
    a.asof_yyyymm,
    a.SUPER_GROUP_NM as Organization,
    a.GROUP_NM as LeadRegion,
    a.REG_NM as Region,
    a.TRTRY_NM as Territory,
    a.MKT_NM as Market,
    'Primary Checking Customer' as Metric,
    (case when ETH_BMG_CD = 1 then 'African American'
         when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then 'Asian'
         when ETH_BMG_CD = 4 then 'Hispanic'
         else 'Other'
        end) as ethnicity,
    generation,
    sum(prim_cnt) as cnt
from primary_summary a,
pop_summary b

where a.SUPER_GROUP_NM = b.SUPER_GROUP_NM
and a.GROUP_NM = b.GROUP_NM
and a.REG_NM = b.REG_NM
and a.TRTRY_NM = b.TRTRY_NM
and a.MKT_NM = b.MKT_NM

group by 1,2,3,4,5,6,7,8,9

union all

select
    a.asof_yyyymm,
    a.SUPER_GROUP_NM as Organization,
    a.GROUP_NM as LeadRegion,
    a.REG_NM as Region,
    a.TRTRY_NM as Territory,
    a.MKT_NM as Market,
    'Store-Based Checking Sales' as Metric,

```

```

        (case when ETH_BMG_CD = 1 then 'African American'
              when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then 'Asian'
              when ETH_BMG_CD = 4 then 'Hispanic'
              else 'Other'
             end) as ethnicity,
        generation,
        sum(all_sales_cnt) as cnt
from ncc_summary a,
pop_summary b

where a.SUPER_GROUP_NM = b.SUPER_GROUP_NM
and a.GROUP_NM = b.GROUP_NM
and a.REG_NM = b.REG_NM
and a.TRTRY_NM = b.TRTRY_NM
and a.MKT_NM = b.MKT_NM

group by 1,2,3,4,5,6,7,8,9

union all

select
  a.asof_yyyymm,
  a.SUPER_GROUP_NM as Organization,
  a.GROUP_NM as LeadRegion,
  a.REG_NM as Region,
  a.TRTRY_NM as Territory,
  a.MKT_NM as Market,
  'New Checking Customer Sales' as Metric,
  (case when ETH_BMG_CD = 1 then 'African American'
        when ETH_BMG_CD in (2,8,9,10,11,12,13,14) then 'Asian'
        when ETH_BMG_CD = 4 then 'Hispanic'
        else 'Other'
       end) as ethnicity,
  generation,
  sum(ncc_cnt) as cnt
from ncc_summary a,
pop_summary b

where a.SUPER_GROUP_NM = b.SUPER_GROUP_NM
and a.GROUP_NM = b.GROUP_NM
and a.REG_NM = b.REG_NM
and a.TRTRY_NM = b.TRTRY_NM
and a.MKT_NM = b.MKT_NM

group by 1,2,3,4,5,6,7,8,9

order by 1,2,3,4,5,6,7,8,9
;

quit;

proc export

```

```
data=metric_final
  dblabel
dbms=xlsx

outfile='/sas/AU48750/BusinessAnalytics/QuarterlyDDA/data/metric_final
.xlsx'
  replace;
run;
```