```
#!/bin/bash


vTable=$1      # This should be the source of the tables you're stepping
               # through taking chunks to compare with the other
               # table.
               # e.g.:  SPDB
               # The other table doesn't matter with respect to (WRT)
               # this running.

me=rsmith
nchunks=100


#  Take one commented out SQL queries below and (elsewhere for clarity here) make it
#  all one string w/o linefeeds.
#  Put that text into variable 'query'.

# ALSO, this script will try to delete temp tables before it creates them, so we
# should create them if they dont' exist when we start this.
# See below.
#
# psql -U $me -d wa_merged -h devdb -c "create table core_spdb_names_part (xyz int) ;"
# psql -U $me -d wa_merged -h devdb -c "create table core_spdb_addrs_part (xyz int) ;"
# psql -U $me -d wa_merged -h devdb -c "create table core_dddb_names_part (xyz int) ;"
# psql -U $me -d wa_merged -h devdb -c "create table core_dddb_addrs_part (xyz int) ;"



# ========================================================================
clear

# -- ------------------------------------------------------
#    Pass in "spdb" for this run

#  create table merged_w_spdb as
#  select cmn.sourceid as merged_id, cspn.sourceid as spdb_id
#  from   core_merged_names          cmn,
#         core_merged_addrs          cma,
#         merged                     m,
#         core_spdb_names_part       cspn,
#         core_spdb_addrs_part       cspa,
#         raw_special_databases      rspd
#  where ((cmn.trim_name = cspn.trim_name)
#    and  (cma.trim_addr = cspa.trim_addr ))
#  OR    ((cmn.trim_name = cspn.trim_name)
#    and  (m.phonenumber = rspd.phone))
#  OR    ((cma.trim_addr = cspa.trim_addr)
#    and  (m.phonenumber = rspd.phone)) ;

```

```
# -- ------------------------------------------------------
#    Pass in "dddb" for this run

#  create table merged_w_dddb as
#  select cmn.sourceid as merged_id, cddn.sourceid as dddb_id
#  from   core_merged_names          cmn,
#         core_merged_addrs          cma,
#         merged                     m,
#         core_dddb_names_part       cddn,
#         core_dddb_addrs_part       cdda,
#         raw_dddb                   rddb
#  where ((cmn.trim_name = cddn.trim_name)
#    and  (cma.trim_addr = cdda.trim_addr ))
#  OR    ((cmn.trim_name = cddn.trim_name)
#    and  (m.phonenumber = rddb.phone))
#  OR    ((cma.trim_addr = cdda.trim_addr)
#    and  (m.phonenumber = rddb.phone)) ;

# -- ------------------------------------------------------
#    Pass in "spdb" for this run

#  create table spdb_w_dddb as
#  select cspn.sourceid as spdb_id, cddn.sourceid as dddb_id
#  from   core_dddb_names           cddn,
#         core_dddb_addrs           cdda,
#         raw_dddb                  rddb
#         core_spdb_names_part      cspn,
#         core_spdb_addrs_part      cspa,
#         raw_special_databases rspd
#  where ((cspn.trim_name = cddn.trim_name)
#    and  (cspa.trim_addr = cdda.trim_addr ))
#  OR    ((cspn.trim_name = cddn.trim_name)
#    and  (rspd.phone     = rddd.phone))
#  OR    ((cspa.trim_addr = cdda.trim_addr)
#    and  (rspd.phone     = rddd.phone)) ;

# -- ------------------------------------------------------


query="create table merged_w_spdb as select cmn.sourceid as merged_id, cspn.sourceid as
  spdb_id  from   core_merged_names     cmn, core_merged_addrs     cma, merged
  m, core_spdb_names       cspn, core_spdb_addrs       cspa, raw_special_databases rspd
  where ((cmn.trim_name = cspn.trim_name) and (cma.trim_addr = cspa.trim_addr))  OR
  ((cmn.trim_name = cspn.trim_name) and (m.phonenumber = rspd.phone))  OR  ((cma.trim_addr
  = cspa.trim_addr) and (m.phonenumber = rspd.phone)) ;"

# ------------------------------------------------------------------------


```

```
vTable=`echo $vTable | tr "[:lower:]" "[:upper:]"`   # Translate lower to uppercase

case $vTable in
"SPDB")
      parting_table="core_spdb_names"  # We just need the #
      ;;
"DDDB")
      parting_table="core_dddb_names"  # We just need the #
      ;;
esac

echo "Determining the # of records in $vTable."
nrows=`psql -U $me -d wa_merged -h devdb -A -t -c "select count(*) from $parting_table ;
  "`
echo -e "Found $nrows records.\n"
rowsperchunck=$((nrows/nchunks))
vOffset=0

while [ $vOffset -le $nrows ]
do

  # --------------------------------------------------------

  case $vTable in
  "SPDB")
        psql -U $me -h devdb -d wa_merged -c "drop table core_spdb_names_part;"
        part_name_query="create table core_spdb_names_part as select * from
          core_spdb_names order by sourceid offset $vOffset limit $rowsperchunk ; "

        psql -U $me -h devdb -d wa_merged -c "drop table core_spdb_addrs_part;"
        part_addr_query="create table core_spdb_addrs_part as select * from
          core_spdb_addrs order by sourceid offset $vOffset limit $rowsperchunk ; "
        ;;

  "DDDB")
        psql -U $me -h devdb -d wa_merged -c "drop table core_dddb_names_part;"
        part_name="create table core_dddb_names_part as select * from core_dddb_names
          order by sourceid offset $vOffset limit $rowsperchunk ; "
        psql -U $me -h devdb -d wa_merged -c "drop table core_dddb_addrs_part;"
        part_addr="create table core_ddb_addrs_part as select * from core_ddb_addrs order
          by sourceid offset $vOffset limit $rowsperchunk ; "
        ;;

  esac

  # --------------------------------------------------------
  t0=`date +"%s"`
  echo -e "\nBegan matching values to get IDs for merge for table '$vTable' at `date '+%D
    %T'`."

```

```
151   echo "SET STATEMENT_TIMEOUT=7200000 ; $query " | psql -U $me -h devdb -d wa_merged -f -
152
153
154   t1=`date +"%s"`
155   echo -e "It took  $((t1-t0)) seconds to process $rowsperchunk rows from $vTable with the
156     'merged' table."
157
158   vOffset=$((vOffset+rowsperchunk))
159
160 done
161
```