

```
1 #!/bin/bash
2
3
4 vTable=$1      # This should be the source of the tables you're stepping
5                # through taking chunks to compare with the other
6                # table.
7                # e.g.: SPDB
8                # The other table doesn't matter with respect to (WRT)
9                # this running.
10
11 me=rsmith
12 nchunks=100
13
14
15 # Take one commented out SQL queries below and (elsewhere for clarity here) make it
16 # all one string w/o linefeeds.
17 # Put that text into variable 'query'.
18
19 # ALSO, this script will try to delete temp tables before it creates them, so we
20 # should create them if they dont' exist when we start this.
21 # See below.
22 #
23 # psql -U $me -d wa_merged -h devdb -c "create table core_spdb_names_part (xyz int) ;"
24 # psql -U $me -d wa_merged -h devdb -c "create table core_spdb_addrs_part (xyz int) ;"
25 # psql -U $me -d wa_merged -h devdb -c "create table core_dddb_names_part (xyz int) ;"
26 # psql -U $me -d wa_merged -h devdb -c "create table core_dddb_addrs_part (xyz int) ;"
27
28
29
30 # =====
31 clear
32
33 # -- =====
34 # Pass in "spdb" for this run
35
36 # create table merged_w_spdb as
37 # select cmn.sourceid as merged_id, cspn.sourceid as spdb_id
38 # from   core_merged_names      cmn,
39 #        core_merged_addrs      cma,
40 #        merged                 m,
41 #        core_spdb_names_part   cspn,
42 #        core_spdb_addrs_part   cspa,
43 #        raw_special_databases  rspd
44 # where ((cmn.trim_name = cspn.trim_name)
45 #        and (cma.trim_addr = cspa.trim_addr ))
46 # OR    ((cmn.trim_name = cspn.trim_name)
47 #        and (m.phonenumber = rspd.phone))
48 # OR    ((cma.trim_addr = cspa.trim_addr)
49 #        and (m.phonenumber = rspd.phone)) ;
50
```

```
51 # -- =====
52 #   Pass in "dddb" for this run
53
54 #   create table merged_w_dddb as
55 #   select cmn.sourceid as merged_id, cddn.sourceid as dddb_id
56 #   from     core_merged_names      cmn,
57 #           core_merged_addrs      cma,
58 #           merged                  m,
59 #           core_dddb_names_part    cddn,
60 #           core_dddb_addrs_part     cdda,
61 #           raw_dddb                rddb
62 #   where ((cmn.trim_name = cddn.trim_name)
63 #         and (cma.trim_addr = cdda.trim_addr ))
64 #   OR     ((cmn.trim_name = cddn.trim_name)
65 #         and (m.phonenumber = rddb.phone))
66 #   OR     ((cma.trim_addr = cdda.trim_addr)
67 #         and (m.phonenumber = rddb.phone)) ;
68
69 # -- =====
70 #   Pass in "spdb" for this run
71
72 #   create table spdb_w_dddb as
73 #   select cspn.sourceid as spdb_id, cddn.sourceid as dddb_id
74 #   from     core_dddb_names      cddn,
75 #           core_dddb_addrs      cdda,
76 #           raw_dddb             rddb
77 #           core_spdb_names_part  cspn,
78 #           core_spdb_addrs_part  cspa,
79 #           raw_special_databases rspd
80 #   where ((cspn.trim_name = cddn.trim_name)
81 #         and (cspa.trim_addr = cdda.trim_addr ))
82 #   OR     ((cspn.trim_name = cddn.trim_name)
83 #         and (rspd.phone      = rddb.phone))
84 #   OR     ((cspa.trim_addr = cdda.trim_addr)
85 #         and (rspd.phone      = rddb.phone)) ;
86
87 # -- =====
88
89
90 query="create table merged_w_spdb as select cmn.sourceid as merged_id, cspn.sourceid as
91 spdb_id from core_merged_names cmn, core_merged_addrs cma, merged
92 m, core_spdb_names cspn, core_spdb_addrs cspa, raw_special_databases rspd
93 where ((cmn.trim_name = cspn.trim_name) and (cma.trim_addr = cspa.trim_addr)) OR
94 ((cmn.trim_name = cspn.trim_name) and (m.phonenumber = rspd.phone)) OR ((cma.trim_addr
95 = cspa.trim_addr) and (m.phonenumber = rspd.phone)) ;"
96
97 # =====
98
99
100
```

```
101 vTable=`echo $vTable | tr "[:lower:]" "[:upper:]"` # Translate lower to uppercase
102
103 case $vTable in
104 "SPDB")
105     parting_table="core_spdb_names" # We just need the #
106     ;;
107 "DDDB")
108     parting_table="core_dddb_names" # We just need the #
109     ;;
110 esac
111
112 echo "Determining the # of records in $vTable."
113 nrows=`psql -U $me -d wa_merged -h devdb -A -t -c "select count(*) from $parting_table ;`
114 ``
115 echo -e "Found $nrows records.\n"
116 rowsperchunk=$((nrows/nchunks))
117 vOffset=0
118
119 while [ $vOffset -le $nrows ]
120 do
121
122     # -----
123
124     case $vTable in
125     "SPDB")
126         psql -U $me -h devdb -d wa_merged -c "drop table core_spdb_names_part;"
127         part_name_query="create table core_spdb_names_part as select * from
128             core_spdb_names order by sourceid offset $vOffset limit $rowsperchunk ; "
129
130         psql -U $me -h devdb -d wa_merged -c "drop table core_spdb_addrs_part;"
131         part_addr_query="create table core_spdb_addrs_part as select * from
132             core_spdb_addrs order by sourceid offset $vOffset limit $rowsperchunk ; "
133         ;;
134
135     "DDDB")
136         psql -U $me -h devdb -d wa_merged -c "drop table core_dddb_names_part;"
137         part_name="create table core_dddb_names_part as select * from core_dddb_names
138             order by sourceid offset $vOffset limit $rowsperchunk ; "
139         psql -U $me -h devdb -d wa_merged -c "drop table core_dddb_addrs_part;"
140         part_addr="create table core_ddb_addrs_part as select * from core_ddb_addrs order
141             by sourceid offset $vOffset limit $rowsperchunk ; "
142         ;;
143
144     esac
145
146     # -----
147     t0=`date +%s`
148     echo -e "\nBegan matching values to get IDs for merge for table '$vTable' at `date +%D
149         %T`."
150
```

```
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
```