Randomisation code for reducing multiple responses
Technical paper

October 2005
Note: This technical paper has been prepared by the Census Evaluations team of Statistics New Zealand. The randomisation code in this paper was developed from the methodology paper Statistics New Zealand (April 2005). “When individual responses exceed input storage – A procedure for unbiased reduction” Technical paper for software developers.

The main procedure is ‘ethnicity_randomisation’ which in turn calls the other three. These stored procedures have been written specifically for Microsoft SQL Server 2000 and may need to be modified for running on other platforms.

Create Proc dbo.ethnicity_randomisation( @batch_nbr int,
                                      @error_msg_text varchar(254)
                                      Output)
As
Begin

-- Business Function : This derivation selects 6 out of 14 ethnicities
-- Input Parameters:
-- @batch_nbr: unique dwelling identifier.
-- Output Parameters: <none>
-- Result Sets Returned: <none>  
-- Return Status:
-- 0 = ok, else @@error is returned and raiserror has been called.  
-- Calls:<none>

Declare
  @error_nbr             int,  -- storing @@error
  @count                int,  -- storing @@rowcount
  @proc_name_text        varchar (30),  -- proc name
  @Person_nbr    int,  -- used to loop through
  @prevPerson_nbr    int           -- used to loop through
Set  nocount on

Select
  @proc_name_text  = 'ethnicity_randomisation'

-- First select the 14 ethnic codes into a holding table for processing
-- This uses a cross join onto a table of 14 records for optimal performance
delete from  ethnic_rand_working
insert into  ethnic_rand_working
select
  i.batch_nbr,
  i.person_nbr,
  num.ethnic_code_nbr,
  case
    when num.ethnic_code_nbr = 1 then ethnic_grp1_code
    when num.ethnic_code_nbr = 2 then ethnic_grp2_code
    when num.ethnic_code_nbr = 3 then ethnic_grp3_code
    when num.ethnic_code_nbr = 4 then ethnic_grp4_code
    when num.ethnic_code_nbr = 5 then ethnic_grp5_code
  end
  i.person_nbr,
  num.ethnic_code_nbr,
when num.ethnic_code_nbr = 6 then ethnic_grp6_code
when num.ethnic_code_nbr = 7 then ethnic_grp7_code
when num.ethnic_code_nbr = 8 then ethnic_grp8_code
when num.ethnic_code_nbr = 9 then ethnic_grp9_code
when num.ethnic_code_nbr = 10 then ethnic_grp10_code
when num.ethnic_code_nbr = 11 then ethnic_grp11_code
when num.ethnic_code_nbr = 12 then ethnic_grp12_code
when num.ethnic_code_nbr = 13 then ethnic_grp13_code
when num.ethnic_code_nbr = 14 then ethnic_grp14_code
end as ethnic_grp_code,

case
when num.ethnic_code_nbr < 9 then null
when num.ethnic_code_nbr = 9 then ethnic_grp9_line_code
when num.ethnic_code_nbr = 10 then ethnic_grp10_line_code
when num.ethnic_code_nbr = 11 then ethnic_grp11_line_code
when num.ethnic_code_nbr = 12 then ethnic_grp12_line_code
when num.ethnic_code_nbr = 13 then ethnic_grp13_line_code
when num.ethnic_code_nbr = 14 then ethnic_grp14_line_code
end as ethnic_grp_line_code,

case
when num.ethnic_code_nbr < 9 then null
when num.ethnic_code_nbr = 9 then ethnic_grp9_auto_code
when num.ethnic_code_nbr = 10 then ethnic_grp10_auto_code
when num.ethnic_code_nbr = 11 then ethnic_grp11_auto_code
when num.ethnic_code_nbr = 12 then ethnic_grp12_auto_code
when num.ethnic_code_nbr = 13 then ethnic_grp13_auto_code
when num.ethnic_code_nbr = 14 then ethnic_grp14_auto_code
end as ethnic_grp_auto_code,

0 as removed_ind,
0 as candidate_to_remove_ind,
null as random_text
from (select batch_nbr, person_nbr,
  ethnic_grp1_code, ethnic_grp2_code, ethnic_grp3_code,
  ethnic_grp4_code, ethnic_grp5_code, ethnic_grp6_code,
  ethnic_grp7_code, ethnic_grp8_code, ethnic_grp9_code,
  ethnic_grp10_code, ethnic_grp11_code, ethnic_grp12_code,
  ethnic_grp13_code, ethnic_grp14_code, ethnic_grp9_line_code,
  ethnic_grp10_line_code,
  ethnic_grp11_line_code,ethnic_grp12_line_code,
  ethnic_grp13_line_code,
  ethnic_grp14_line_code,ethnic_grp9_auto_code,
  ethnic_grp10_auto_code, ethnic_grp11_auto_code,
  ethnic_grp12_auto_code, ethnic_grp13_auto_code,
  ethnic_grp14_auto_code
from dbo.individual
where batch_nbr = @batch_nbr ) i
cross join dbo.ethnic_rand_number num

select @error_nbr = @@error

if @error_nbr <> 0
  begin
    select @error_msg_text = @proc_name_text + ' Step 1'
    return @error_nbr
  end

-- Remove any duplicates and set the corresponding record to 99999

update dbo.ethnic_rand_working
set ethnic_grp_code = 99999
from dbo.ethnic_rand_working qryAll
inner join (select batch_nbr, person_nbr, ethnic_grp_code
from dbo.ethnic_rand_working
where batch_nbr = @batch_nbr
and ethnic_grp_code <> 99999
group by batch_nbr, person_nbr, ethnic_grp_code
having count(*) > 1) qryDup
on qryAll.batch_nbr = qryDup.batch_nbr
and qryAll.person_nbr = qryDup.person_nbr
and qryAll.ethnic_grp_code = qryDup.ethnic_grp_code
left join (select batch_nbr, person_nbr,
ethnic_grp_code, min(ethnic_code_nbr) as
ethnic_code_nbr
from dbo.ethnic_rand_working
where batch_nbr = @batch_nbr
and ethnic_grp_code <> 99999
group by batch_nbr, person_nbr, ethnic_grp_code
having count(*) > 1) qryDupToKeep
on qryAll.batch_nbr = qryDupToKeep.batch_nbr
and qryAll.person_nbr = qryDupToKeep.person_nbr
and qryAll.ethnic_code_nbr = qryDupToKeep.ethnic_code_nbr
where qryAll.batch_nbr = @batch_nbr
and qryDupToKeep.batch_nbr is null
and qryAll.ethnic_grp_code <> 99999

if @error_nbr <> 0
begin
select @error_msg_text = @proc_name_text + ' Step 2'
return @error_nbr
end

-- Loop through all the individuals in the batch and process them separately
select @Person_nbr = 0
while 1=1
begin
select @prevPerson_nbr = @Person_nbr
select top 1 @Person_nbr = person_nbr
from ethnic_rand_working
where person_nbr > @prevPerson_nbr
order by person_nbr

select @error_nbr = @error, @count = @rowcount
if @error_nbr = 0 and @count = 1
begin

-- Reduce the individual's ethnicities from 14 to 6
exec @error_nbr = ethnicity_randomisation_person
@person_nbr,
@error_msg_text, 6
if @error_nbr <> 0
return @error_nbr

-- Reduce the individual's ethnicities from 6 to 3
exec @error_nbr = ethnicity_randomisation_person
@person_nbr,
@error_msg_text, 3
if @error_nbr <> 0
return @error_nbr

end
else if @error_nbr  = 0 and @count  = 0
break
else if @error_nbr <> 0
begin
select @error_msg_text = @proc_name_text + ' Step 3'
return @error_nbr
end
end

end

-- This following PROC is for person

Create Proc dbo.ethnicity_randomisation_person (@person_nbr  int,
@error_msg_text  varchar(254)
Output,
@ethnicities_to_leave  smallint )
As
Begin

-- Procedure Name: ethnicity_randomisation_person
-- Business Function : This proc is called by ethnicity_randomisation
-- It will select 6 out of 14 ethnicity codes for a given person

Declare
@error_nbr    int ,         -- storing @@error
@count_nbr   int ,         -- storing @@rowcount
@proc_name_text      varchar (30), -- proc name
@remaining_nbr  int ,         -- number of codes remaining
@remove_nbr   int           -- number of codes to remove

Set nocount on

Select
@proc_name_text  = ethnicity_randomisation_person'

if @ethnicities_to_leave  = 6
Select @remaining_nbr  = 14 -- start with 14 ethnicities
else if @ethnicities_to_leave  = 3
Select @remaining_nbr  = 6 -- start with 6 ethnicities

update ethnic_rand_working
set candidate_to_remove_ind = 1
where person_nbr = @person_nbr
and removed_ind = 0
and ethnic_grp_code = '99999'

select @count_nbr  = @@rowcount

if @ethnicities_to_leave  = 6 and @count_nbr >= 8
begin
-- most individuals will have less than 6 ethnicities
-- so remove 8 of 99999
update ethnic_rand_working
set removed_ind = 1
from ethnic_rand_working
inner join (select top 8 person_nbr, ethnic_code_nbr
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0
and ethnic_grp_code = '99999'
order by newid() ) qryRemove
on ethnic_rand_working.person_nbr = qryRemove.person_nbr
and ethnic_rand_working.ethnic_code_nbr =
qryRemove.ethnic_code_nbr
    select @error_nbr = @@error
end
else if @ethnicities_to_leave = 3 and @count_nbr >= 3
begin
    -- most individuals will have less than 3 ethnicities so remove 3
    of
    -- 99999
    update ethnic_rand_working
    set removed_ind = 1
from ethnic_rand_working
inner join (select top 3 person_nbr, ethnic_code_nbr
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0
and ethnic_grp_code = '99999'
order by newid() ) qryRemove
on ethnic_rand_working.person_nbr = qryRemove.person_nbr
and ethnic_rand_working.ethnic_code_nbr =
qryRemove.ethnic_code_nbr
    select @error_nbr = @@error
end
else if @count_nbr < 0
begin
    set @remove_nbr = @remaining_nbr - @ethnicities_to_leave
    exec @error_nbr = ethnicity_randomisation_remove @person_nbr,
@error_msg_text,
@remove_nbr
end
else if @error_nbr <> 0
    return @error_nbr
select @remaining_nbr = count(*)
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0
if @remaining_nbr = @ethnicities_to_leave
    GOTO Update_Individual

---------------------------------------------------------------------
--- Step 2) Removing residual Responses i.e. 9****
---------------------------------------------------------------------


update ethnic_rand_working
set candidate_to_remove_ind = 1
where person_nbr = @person_nbr
and removed_ind = 0
and ethnic_grp_code like '9%'

select @count_nbr = @@rowcount

if @count_nbr <> 0
begin
    set @remove_nbr = @remaining_nbr - @ethnicities_to_leave
    exec @error_nbr = ethnicity_randomisation_remove
            @person_nbr, @error_msg_text, @remove_nbr
end

if @error_nbr <> 0
return @error_nbr

select @remaining_nbr = count (*)
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0

if @remaining_nbr = @ethnicities_to_leave
GOTO Update_Individual

-------------------------------------------------------------------------------
----
-- Step 3) Removing 'non-informative' real responses
--
-- i.e. Remove a less specific ethnicities when a
-- more specific ethnicity is also selected
--
-- e.g. In the selection 40000 Asian nfd, 42100 Chinese nfd & 42111 Hong Kong Chinese
-- the first two are less specific answers and can
-- therefore be removed.
-------------------------------------------------------------------------------
----
-- remove parents (p) of the form x0000 when there is
-- a child (c) with p < c < p+10000
update ethnic_rand_working
set candidate_to_remove_ind = 1
from ethnic_rand_working ec
inner join (select batch_nbr, person_nbr, ethnic_code_nbr, ethnic_grp_code as parent_grp_code, removed_ind
        from ethnic_rand_working
        where ethnic_grp_code like '%0000' ) qryParent
        on ec.person_nbr = qryParent.person_nbr
        and ec.ethnic_code_nbr = qryParent.ethnic_code_nbr
inner join (select batch_nbr, person_nbr,
left(ethnic_grp_code, 1) + '0000' as parent_grp_code,
    ethnic_grp_code
from ethnic_rand_working
where ethnic_grp_code not like '%0000'
  on qryParent.person_nbr = qryChild.person_nbr
  and qryParent.parent_grp_code = qryChild.parent_grp_code
and qryParent.person_nbr = @person_nbr
and qryParent.removed_ind = 0

select @count_nbr = @@rowcount
if (@count_nbr <> 0)
begin
  set @remove_nbr = @remaining_nbr - @ethnicities_to_leave
  exec @error_nbr = ethnicity_randomisation_remove
    @person_nbr,
    @error_msg_text,
    @remove_nbr
end

if @error_nbr <> 0
  return @error_nbr

select @remaining_nbr = count(*)
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0

if @remaining_nbr = @ethnicities_to_leave
  GOTO Update_Individual

-- remove parents (p) of the form xy000 when
-- there is a child (c) with p < c < p+1000
update ethnic_rand_working
set candidate_to_remove_ind = 1
from ethnic_rand_working ec
inner join (select batch_nbr,
    person_nbr,
    ethnic_code_nbr,
    ethnic_grp_code as parent_grp_code,
    removed_ind
  from ethnic_rand_working
  where ethnic_grp_code like '%000'
    on ec.person_nbr = qryParent.person_nbr
    and ec.ethnic_code_nbr = qryParent.ethnic_code_nbr
inner join (select batch_nbr,
    person_nbr,
    left(ethnic_grp_code, 2) + '000' as parent_grp_code,
    ethnic_grp_code
  from ethnic_rand_working
  where ethnic_grp_code not like '%000'
    on qryParent.person_nbr = qryChild.person_nbr
    and qryParent.parent_grp_code = qryChild.parent_grp_code
    where qryParent.person_nbr = @person_nbr
    and qryParent.removed_ind = 0
select @count_nbr = @@rowcount
if @count_nbr <> 0
begin
set @remove_nbr = @remaining_nbr - @ethnicities_to_leave
exec @error_nbr = ethnicity_randomisation_remove @person_nbr,
       @error_msg_text,
       @remove_nbr
end

if @error_nbr <> 0
return @error_nbr

select @remaining_nbr = count(*)
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0

if @remaining_nbr = @ethnicities_to_leave
    GOTO Update_Individual

-- Remove parents (p) of the form xyz00 when there
-- is a child (c) with p < c < p+1000
update ethnic_rand_working
set candidate_to_remove_ind = 1
from ethnic_rand_working ec
inner join (select batch_nbr,
            person_nbr,
            ethnic_code_nbr,
            ethnic_grp_code as parent_grp_code,
            removed_ind
     from ethnic_rand_working
     where ethnic_grp_code like '%00' ) qryParent
on ec.person_nbr = qryParent.person_nbr
and ec.ethnic_code_nbr = qryParent.ethnic_code_nbr
inner join (select batch_nbr,
            person_nbr,
            left (ethnic_grp_code, 3) + '00' as parent_grp_code,
            ethnic_grp_code
     from ethnic_rand_working
     where ethnic_grp_code not like '%00' ) qryChild
on qryParent.person_nbr = qryChild.person_nbr
and qryParent.parent_grp_code = qryChild.parent_grp_code
where qryParent.person_nbr = @person_nbr
and qryParent.removed_ind = 0

select @count_nbr = @@rowcount
if @count_nbr <> 0
begin
set @remove_nbr = @remaining_nbr - @ethnicities_to_leave
exec @error_nbr = ethnicity_randomisation_remove @person_nbr,
       @error_msg_text,
       @remove_nbr
end

if @error_nbr <> 0
return @error_nbr

select @remaining_nbr = count(*)
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0

if @remaining_nbr = @ethnicities_to_leave
   GOTO Update_Individual

----
-- Step 4) Removing 'informative' real responses
--     i.e. Identify ethnicities as candidates for
--     removal where similar ethnicities are present
--     randomly remove one of those candidates and
--     repeat as necessary
----

declare @counter int
set @counter = 1

while (@remaining_nbr > @ethnicities_to_leave and @counter <= 14)
begin
  Call proc to identify candidates and remove 1 ethnicity
  exec
  @error_nbr=ethnicity_randomisation_person_identify_candidate
     @person_nbr,
     @error_msg_text,
     @ethnicities_to_leave

  if @error_nbr <> 0
     return @error_nbr

  select @remaining_nbr = count(*)
  from ethnic_rand_working
  where person_nbr = @person_nbr
  and removed_ind = 0

  if @counter = 14
     RAISERROR ('ethnicity_randomisation_person Step 1', 16, 1)
     -- Failed to reduce ethnicities
     set @counter = @counter + 1
  end

-- Enter the remaining codes into the appropriate fields
----

Update_Individual:

declare @eth_nbr1 int,
     @eth_nbr2 int,
     @eth_nbr3 int,
     @eth_nbr4 int,
     @eth_nbr5 int,
     @eth_nbr6 int

select top 1 @eth_nbr1 = ethnic_code_nbr
from ethnic_rand_working
where removed_ind = 0 and person_nbr = @person_nbr
order by ethnic_grp_code

select top 1 @eth_nbr2 = ethnic_code_nbr
from ethnic_rand_working
where removed_ind = 0 and person_nbr = @person_nbr
    and ethnic_code_nbr not in (@eth_nbr1)
order by ethnic_grp_code

select top 1 @eth_nbr3 = ethnic_code_nbr
from ethnic_rand_working
where removed_ind = 0 and person_nbr = @person_nbr
    and ethnic_code_nbr not in (@eth_nbr1, @eth_nbr2)
order by ethnic_grp_code

if @ethnicities_to_leave = 6
begin
select top 1 @eth_nbr4 = ethnic_code_nbr
from ethnic_rand_working
where removed_ind = 0 and person_nbr = @person_nbr
    and ethnic_code_nbr not in (@eth_nbr1, @eth_nbr2, @eth_nbr3)
order by ethnic_grp_code

select top 1 @eth_nbr5 = ethnic_code_nbr
from ethnic_rand_working
where removed_ind = 0 and person_nbr = @person_nbr
    and ethnic_code_nbr not in (@eth_nbr1, @eth_nbr2, @eth_nbr3, @eth_nbr4)
order by ethnic_grp_code

select top 1 @eth_nbr6 = ethnic_code_nbr
from ethnic_rand_working
where removed_ind = 0 and person_nbr = @person_nbr
    and ethnic_code_nbr not in (@eth_nbr1, @eth_nbr2, @eth_nbr3, @eth_nbr4, @eth_nbr5)
order by ethnic_grp_code

update dbo.individual_v
set
    ethnic_rand6_grp1_code = qryEth1.ethnic_grp_code,
    ethnic_rand6_grp1_line_code = qryEth1.ethnic_grp_line_code,
    ethnic_rand6_grp1_auto_code = qryEth1.ethnic_grp_auto_code,
    ethnic_rand6_grp2_code = qryEth2.ethnic_grp_code,
    ethnic_rand6_grp2_line_code = qryEth2.ethnic_grp_line_code,
    ethnic_rand6_grp2_auto_code = qryEth2.ethnic_grp_auto_code,
    ethnic_rand6_grp3_code = qryEth3.ethnic_grp_code,
    ethnic_rand6_grp3_line_code = qryEth3.ethnic_grp_line_code,
    ethnic_rand6_grp3_auto_code = qryEth3.ethnic_grp_auto_code,
    ethnic_rand6_grp4_code = qryEth4.ethnic_grp_code,
    ethnic_rand6_grp4_line_code = qryEth4.ethnic_grp_line_code,
    ethnic_rand6_grp4_auto_code = qryEth4.ethnic_grp_auto_code,
    ethnic_rand6_grp5_code = qryEth5.ethnic_grp_code,
    ethnic_rand6_grp5_line_code = qryEth5.ethnic_grp_line_code,
    ethnic_rand6_grp5_auto_code = qryEth5.ethnic_grp_auto_code,
    ethnic_rand6_grp6_code = qryEth6.ethnic_grp_code,
    ethnic_rand6_grp6_line_code = qryEth6.ethnic_grp_line_code,
    ethnic_rand6_grp6_auto_code = qryEth6.ethnic_grp_auto_code
from dbo.individual_v
    inner join (select person_nbr, ethnic_grp_code,
    ethnic_grp_line_code,
    ethnic_grp_auto_code
    from ethnic_rand_working
}
```sql
WHERE person_nbr = @person_nbr
    AND ethnic_code_nbr = @eth_nbr1) qryEth1
ON i.person_nbr = qryEth1.person_nbr
INNER JOIN (SELECT person_nbr, ethnic_grp_code,
    ethnic_grp_line_code, ethnic_grp_auto_code
FROM ethnic_rand_working
    WHERE person_nbr = @person_nbr
    AND ethnic_code_nbr = @eth_nbr2) qryEth2
ON i.person_nbr = qryEth2.person_nbr
INNER JOIN (SELECT person_nbr, ethnic_grp_code,
    ethnic_grp_line_code, ethnic_grp_auto_code
FROM ethnic_rand_working
    WHERE person_nbr = @person_nbr
    AND ethnic_code_nbr = @eth_nbr3) qryEth3
ON i.person_nbr = qryEth3.person_nbr
INNER JOIN (SELECT person_nbr, ethnic_grp_code,
    ethnic_grp_line_code, ethnic_grp_auto_code
FROM ethnic_rand_working
    WHERE person_nbr = @person_nbr
    AND ethnic_code_nbr = @eth_nbr4) qryEth4
ON i.person_nbr = qryEth4.person_nbr
INNER JOIN (SELECT person_nbr, ethnic_grp_code,
    ethnic_grp_line_code, ethnic_grp_auto_code
FROM ethnic_rand_working
    WHERE person_nbr = @person_nbr
    AND ethnic_code_nbr = @eth_nbr5) qryEth5
ON i.person_nbr = qryEth5.person_nbr
INNER JOIN (SELECT person_nbr, ethnic_grp_code,
    ethnic_grp_line_code, ethnic_grp_auto_code
FROM ethnic_rand_working
    WHERE person_nbr = @person_nbr
    AND ethnic_code_nbr = @eth_nbr6) qryEth6
END
ELSE
    IF @ethnicities_to_leave = 3
    BEGIN
        UPDATE dbo.individual_v
        SET
            ethnic_rand3_grp1_code   = qryEth1.ethnic_grp_code,
            ethnic_rand3_grp1_line_code = qryEth1.ethnic_grp_line_code,
            ethnic_rand3_grp1_auto_code = qryEth1.ethnic_grp_auto_code,
            ethnic_rand3_grp2_code   = qryEth2.ethnic_grp_code,
            ethnic_rand3_grp2_line_code = qryEth2.ethnic_grp_line_code,
            ethnic_rand3_grp2_auto_code = qryEth2.ethnic_grp_auto_code,
            ethnic_rand3_grp3_code   = qryEth3.ethnic_grp_code,
            ethnic_rand3_grp3_line_code = qryEth3.ethnic_grp_line_code,
            ethnic_rand3_grp3_auto_code = qryEth3.ethnic_grp_auto_code
        FROM dbo.individual_v i
        INNER JOIN (SELECT person_nbr, ethnic_grp_code,
            ethnic_grp_line_code, ethnic_grp_auto_code
        FROM ethnic_rand_working
            WHERE person_nbr = @person_nbr
            AND ethnic_code_nbr = @eth_nbr1) qryEth1
        ON i.person_nbr = qryEth1.person_nbr
        INNER JOIN (SELECT person_nbr, ethnic_grp_code,
            ethnic_grp_line_code, ethnic_grp_auto_code
        FROM ethnic_rand_working
            WHERE person_nbr = @person_nbr
            AND ethnic_code_nbr = @eth_nbr2) qryEth2
        ON i.person_nbr = qryEth2.person_nbr
        INNER JOIN (SELECT person_nbr, ethnic_grp_code,
            ethnic_grp_line_code, ethnic_grp_auto_code
        FROM ethnic_rand_working
            WHERE person_nbr = @person_nbr
            AND ethnic_code_nbr = @eth_nbr3) qryEth3
        ON i.person_nbr = qryEth3.person_nbr
        INNER JOIN (SELECT person_nbr, ethnic_grp_code,
            ethnic_grp_line_code, ethnic_grp_auto_code
        FROM ethnic_rand_working
            WHERE person_nbr = @person_nbr
            AND ethnic_code_nbr = @eth_nbr4) qryEth4
        ON i.person_nbr = qryEth4.person_nbr
        INNER JOIN (SELECT person_nbr, ethnic_grp_code,
            ethnic_grp_line_code, ethnic_grp_auto_code
        FROM ethnic_rand_working
            WHERE person_nbr = @person_nbr
            AND ethnic_code_nbr = @eth_nbr5) qryEth5
        ON i.person_nbr = qryEth5.person_nbr
        INNER JOIN (SELECT person_nbr, ethnic_grp_code,
            ethnic_grp_line_code, ethnic_grp_auto_code
        FROM ethnic_rand_working
            WHERE person_nbr = @person_nbr
            AND ethnic_code_nbr = @eth_nbr6) qryEth6
    END
```

and ethnic_code_nbr = @eth_nbr2) qryEth2
on i.person_nbr = qryEth2.person_nbr
inner join (select person_nbr, ethnic_grp_code,
               ethnic_grp_line_code, ethnic_grp_auto_code
               from ethnic_rand_working
               where person_nbr = @person_nbr
               and ethnic_code_nbr = @eth_nbr3) qryEth3
on i.person_nbr = qryEth3.person_nbr
end
end

-- This following procedures is to identify a candidate
Create Proc dbo.ethnicity_randomisation_person_identify_candidate (   @person_nbr int ,
                @error_msg_text  varchar (254)
Output ,
                @ethnicities_to_leave smallint )
As
Begin
   -- Procedure Name: ethnicity_randomisation_person_identify_candidate
   -- Business Function : This proc is called by
   -- ethnicity_randomisation_person
   -- it is step 4 of the process and used to identify candidates at a certain -- level in the hierarchy of ethnicities
Declare
   @error_nbr            int ,         -- storing @@error
   @count_nbr            int ,         -- storing @@rowcount
   @proc_name_text       varchar (30), -- proc name
   @distinct_response_nbr  int        -- no of distinct responses
at a
   -- given level
Set nocount on
Select
   @proc_name_text  = 'ethnicity_randomisation_person_identify_candidate'
--------------------------------------------------------------
-- Remove real responses at 'level 1'
-- if there are more responses than required at this
-- level then remove one of the ones that share a parent
--------------------------------------------------------------
select @distinct_response_nbr = count ( distinct left (ethnic_grp_code, 1))
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0
group by batch_nbr, person_nbr
if @distinct_response_nbr <= @ethnicities_to_leave
   GOTO Level2
update ethnic_rand_working  --at level 1 'all parents = 0' hence all
   --candidates
set candidate_to_remove_ind = 1
from ethnic_rand_working ec
where person_nbr = @person_nbr
and removed_ind = 0

select @count_nbr = @@rowcount
if @count_nbr <> 0
    exec @error_nbr = ethnicity_randomisation_remove
        @person_nbr,
        @error_msg_text,
        1
return @error_nbr --only remove one ethnicity at a time so return to
calling
    -- proc on success or fail

-- Remove real responses at 'level 2'
-- if there are more responses than required at this
-- level then remove one of the ones that share a parent

Level2:

select @distinct_response_nbr = count(distinct left(ethnic_grp_code, 2))
from ethnic_rand_working
where person_nbr = @person_nbr
and removed_ind = 0
group by batch_nbr, person_nbr
if @distinct_response_nbr <= @ethnicities_to_leave
    GOTO Level3
update ethnic_rand_working
set candidate_to_remove_ind = 1
from ethnic_rand_working ec
inner join (select batch_nbr,
            person_nbr,
            left(ethnic_grp_code, 1) as level_1,
            count(ethnic_code_nbr) as noEthnicities
        from ethnic_rand_working
        where person_nbr = @person_nbr
        and removed_ind = 0
        group by batch_nbr,
                person_nbr,
                left(ethnic_grp_code, 1)
        having count(ethnic_code_nbr) > 1 ) qryCandidates
on ec.person_nbr = qryCandidates.person_nbr
and left(ec.ethnic_grp_code, 1) = qryCandidates.level_1
where ec.removed_ind = 0
select @count_nbr = @@rowcount
if @count_nbr <> 0
    exec @error_nbr = ethnicity_randomisation_remove
        @person_nbr,
        @error_msg_text,
        1

return @error_nbr  -- only remove one ethnicity at a time so return to
calling
    -- proc on success or fail

-- Remove real responses at 'level 3'
-- if there are more responses than required at this
-- level then remove one of the ones that share a parent
-----------------------------------------------

Level3:

select  @distinct_response_nbr  = count ( distinct  left (ethnic_grp_code, 3))
from  ethnic_rand_working
where  person_nbr = @person_nbr
and  removed_ind = 0
group by  batch_nbr, person_nbr

if  @distinct_response_nbr  <=  @ethnicities_to_leave
    GOTO  Level4

update  ethnic_rand_working
set  candidate_to_remove_ind  =  1
from  ethnic_rand_working  ec
inner join  ( select
    batch_nbr,
    person_nbr,
    left (ethnic_grp_code, 2)  as  level_2,
    count (ethnic_code_nbr)  as  noEthnicities
from  ethnic_rand_working
where  person_nbr = @person_nbr
and  removed_ind  =  0
group by  batch_nbr,
    person_nbr,
    left (ethnic_grp_code, 2)
    having  count(ethnic_code_nbr)  >  1  ) qryCandidates
on   ec.person_nbr = qryCandidates.person_nbr
and  left(ec.ethnic_grp_code, 2) = qryCandidates.level_2
where  ec.removed_ind  =  0

select  @count_nbr  = @@rowcount

if  @count_nbr  <>  0
    exec  @error_nbr  =  ethnicity_randomisation_remove
    @person_nbr,
    @error_msg_text,
    1

return  @error_nbr  -- only remove one ethnicity at a time so return to
calling
    -- proc on success or fail

-- Remove real responses at 'level 4'
-- remove one of the ones that share a parent
-- (there nmust be more responses than required as the proc has been
called)
-----------------------------------------------
update ethnic_rand_working
set candidate_to_remove_ind = 1
from ethnic_rand_working ec
inner join ( select batch_nbr,
            person_nbr,
            left(ethnic_grp_code, 3) as level_3,
            count(ethnic_code_nbr) as noEthnicities
        from ethnic_rand_working
        where person_nbr = @person_nbr
        and removed_ind = 0
        group by batch_nbr, person_nbr, left(ethnic_grp_code, 3)
        having count(ethnic_code_nbr) > 1 ) qryCandidates
on ec.person_nbr = qryCandidates.person_nbr
and left(ec.ethnic_grp_code, 3) = qryCandidates.level_3
where ec.removed_ind = 0

select @count_nbr = @@rowcount
if @count_nbr <> 0
  exec @error_nbr = ethnicity_randomisation_remove
    @person_nbr,
    @error_msg_text,
    1
return @error_nbr --only remove one ethnicity at a time so return to calling
  -- proc on success or fail
end

Create Proc dbo.ethnicity_randomisation_remove ( 
  @person_nbr int,
  @error_msg_text varchar (254)
Output,
  @remove_nbr int )
As
  Begin
    -- Procedure Name: ethnicity_randomisation_remove
    -- Business Function : This proc is called by
    -- ethnicity_randomisation_person &
    -- ethnicity_randomisation_person_identify_candidate
    -- which determine ethnicities require reducing and set
candidate_to_remove --= 1
    -- this proc will randomly remove a given number of those candidates
    -- by setting removed = 1 in the holding table (derivation096_ethnic_codes)
    Declare
      @error_nbr int, -- storing @error
      @proc_name_text varchar(30) -- the proc name
    Set nocount on
    Select
      @proc_name_text = 'ethnicity_randomisation_remove'
/* assign every candidate a random number
This is not as easy as it seems.
1. Rand() does produce a random number but 'set candidate_to_remove_ind=Rand()' would give all candidates the same random number.
2. Using a cursor to step through and assign each row a different Rand() is horrendously slow.
3. It is possible to use Rand(seed), but the same seed will always give the same random number, and the random numbers are in the same order as the seed so those with high ethnic_code_nbr would tend to have high random numbers.
4. newID() produces a new unique identifier at random, which can then be used to order the results. /*

update ethnic_rand_working
set random_text =
  case when candidate_to_remove_ind = 1
    then cast(newid() as varchar(36))
  else null
end
where person_nbr = @person_nbr

if @error_nbr <> 0
begin
  select @error_msg_text = @proc_name_text + ' Step 1'
  return @error_nbr
end

-- Set those with the lowest random number as removed
set rowcount @remove_nbr

update ethnic_rand_working
set removed_ind = 1
from ethnic_rand_working
inner join (select top 100 percent *
  from ethnic_rand_working
  where person_nbr = @person_nbr
  and random_text is not null
  order by random_text) ecOrdered
  on ethnic_rand_working.person_nbr = ecOrdered.person_nbr
and ethnic_rand_working.ethnic_code_nbr = ecOrdered.ethnic_code_nbr

select @error_nbr = @@error
set rowcount 0
if @error_nbr <> 0
begin
  select @error_msg_text = @proc_name_text + ' Step 2'
  return @error_nbr
end

-- reset data
update ethnic_rand_working
set candidate_to_remove_ind = 0, random_text = null
where person_nbr = @person_nbr

end