# **Loading the Personson Table**

In order to load the Integration.Person table the following stored procedure has been designed to group Transact-SQL statements into a logical unit.

**Stored procedure:** pstg.Load\_PersonTable

The Load\_PersonTable is a procedure which call 3 other procedures in order to load the Integration.Person table.

Those 3 procedures go from first truncating the base tables which feed the Person Table to loading them

again before starting to load the main table though different defined criteria.

* pstg.Truncate\_Basetables() - procedure which truncate the base tables (**basehbt and basecontact** )
* pstg.Load\_Basetables() - procedure which loads the base tables
* pstg.Load\_AllPersonCriteria() – procedure which load the Person table from the already populated base tables

**Step to run the pstg.Load\_PersonTable procedure**

**CREATE OR REPLACE PROCEDURE Load\_PersonTable()**

**AS $$**

**BEGIN**

**call pstg.Truncate\_Basetables();**

**call pstg.Load\_Basetables();**

**call pstg.Load\_AllPersonCriteria();**

**END;**

**$$ LANGUAGE plpgsql**

**;**

In order to run the run the pstg.Load\_PersonTable procedure the following execute the following statement:

**Call Load\_PersonTable ();**

Once the above statement is executed the following will happen behind the scenes:

* call pstg.Truncate\_Basetables();

CREATE OR REPLACE PROCEDURE pstg.Truncate\_Basetables()

AS $$

BEGIN

delete from pstg.basehbt;

delete from pstg.basecontact;

END;

$$ LANGUAGE plpgsql;

* call pstg.Load\_Basetables();

CREATE OR REPLACE PROCEDURE pstg.Load\_Basetables()

AS $$

BEGIN

----Pull everyone from SF contact who is not already in the person table, excluding test records

insert into pstg.basecontact(

SELECT DISTINCT

contactid,

student\_id,

/\*Logic : firstname = regexp\_replace((upper(firstname::text)), '[^ a-zA-Z]', ' ')\*/

regexp\_replace((upper(firstname::text)), '[^ a-zA-Z]', ' ') AS firstname,

/\*Logic : lastname = regexp\_replace((upper(lastname::text)), '[^ a-zA-Z]', ' ')\*/

regexp\_replace((upper(lastname::text)), '[^ a-zA-Z]', ' ') AS lastname,

social\_security\_number\_last4,

birthdate,

ltrim(rtrim(COALESCE(

CASE

WHEN "replace"("replace"("replace"("replace"("replace"(phone::text, '-'::character varying::text, ''::character varying::text), '('::character varying::text, ''::character varying::text), ')'::character varying::text, ''::character varying::text), ' '::character varying::text, ''::character varying::text), '[^0-9]'::character varying::text, ''::character varying::text) = ''::character varying::text THEN NULL::character varying::text

ELSE "replace"("replace"("replace"("replace"("replace"(phone::text, '-'::character varying::text, ''::character varying::text), '('::character varying::text, ''::character varying::text), ')'::character varying::text, ''::character varying::text), ' '::character varying::text, ''::character varying::text), '[^0-9]'::character varying::text, ''::character varying::text)END, '0'::character varying::text)))

AS phone,

ltrim(rtrim(COALESCE(CASE WHEN "replace"("replace"("replace"("replace"("replace"(mobilephone::text, '-'::character varying::text, ''::character varying::text), '('::character varying::text, ''::character varying::text), ')'::character varying::text, ''::character varying::text), ' '::character varying::text, ''::character varying::text), '[^0-9]'::character varying::text, ''::character varying::text) = ''::character varying::text THEN NULL::character varying::text

ELSE "replace"("replace"("replace"("replace"("replace"(mobilephone::text, '-'::character varying::text, ''::character varying::text), '('::character varying::text, ''::character varying::text), ')'::character varying::text, ''::character varying::text), ' '::character varying::text, ''::character varying::text), '[^0-9]'::character varying::text, ''::character varying::text)END, '0'::character varying::text)))

AS mobilephone,

ltrim(rtrim(COALESCE(

CASE

WHEN "replace"("replace"("replace"("replace"("replace"(homephone::text, '-'::character varying::text, ''::character varying::text), '('::character varying::text, ''::character varying::text), ')'::character varying::text, ''::character varying::text), ' '::character varying::text, ''::character varying::text), '[^0-9]'::character varying::text, ''::character varying::text) = ''::character varying::text THEN NULL::character varying::text

ELSE "replace"("replace"("replace"("replace"("replace"(homephone::text, '-'::character varying::text, ''::character varying::text), '('::character varying::text, ''::character varying::text), ')'::character varying::text, ''::character varying::text), ' '::character varying::text, ''::character varying::text), '[^0-9]'::character varying::text, ''::character varying::text) END, '0'::character varying::text)))

AS homephone,

/\*Logic : dshs\_provider\_id = substring(dshs\_provider\_id::text, 1, 7)\*/

"substring"(dshs\_provider\_id::text, 1, 7) AS dshs\_provider\_id,

mailingpostalcode, status,

/\*Logic : birthdate\_age = ((datediff(year, birthdate, sysdate))-1)\*/

date\_part(y,sysdate)-date\_part(y,birthdate::date) as birthdate\_age ,

---((datediff(year, birthdate, sysdate))-1) as birthdate\_age,

/\*Logic : mailingstreet = upper(mailingstreet)\*/

upper(mailingstreet) AS mailingstreet,

middle\_initial

FROM acquisition.sf\_contact c

where not exists (select 1 from pstg.person t where c.contactid= t.contactid)

/\*Logic : social\_security\_number\_last4 <> '', 0\*/

--and social\_security\_number\_last4 <>''

/\*added this logic\*/

--and cast(social\_security\_number\_last4 as varchar)<>'0'

/\*Logic : firstname <> '%\*%', '%@%', '%T\*ST%', 'TEST', ''\*/

and firstname not like '%\*%'

and firstname not like '%@%'

and upper(firstname) not like 'TEST'

and upper(firstname) not like '%T\*ST%'

/\*added this logic\*/

and firstname<>''

/\*Logic : lastname <> '%\*%', '%@%', '%T\*ST%', 'TEST',''\*/

and lastname not like '%\*%'

and lastname not like '%@%'

and upper(lastname) not like '%T\*ST%'

and upper(lastname) not like 'TEST'

/\*added this logic\*/

and lastname<>''

/\*Logic : upper(status) <> 'DUPLICATE'\*/

and upper(status) not like 'DUPLICATE'

and status <>''

);

-- Pull everyone from hbt who is not in the person table, and put them in basehbt

insert into pstg.basehbt(

select

kn\_key\_name,

/\*prt\_first = regexp\_replace((upper(prt\_first)), '[^ a-zA-Z]', ' ')\*/

regexp\_replace((upper(prt\_first)), '[^ a-zA-Z]', ' ') as prt\_first,

/\*prt\_last = regexp\_replace((upper(prt\_last)), '[^ a-zA-Z]', ' ')\*/

regexp\_replace((upper(prt\_last)), '[^ a-zA-Z]', ' ') as prt\_last,

/\*prt\_ss\_nbr\_last4 = right(prt\_ss\_nbr, 4)\*/

right(prt\_ss\_nbr, 4) as prt\_ss\_nbr\_last4,

prt\_birth\_date,

prt\_employer\_code,

prt\_telephone\_nbr,

/\*prt\_zip\_code = cast(prt\_zip\_code as varchar(20))\*/

cast(prt\_zip\_code as varchar(20)) as prt\_zip\_code ,

prt\_ss\_nbr,

/\*zen\_age = ((datediff(year, prt\_birth\_date, sysdate))-1)\*/

date\_part(y,sysdate)-date\_part(y,prt\_birth\_date::date) as zen\_age ,

--used date part to calculate the difference

---((datediff(year, prt\_birth\_date, sysdate))-1) as zen\_age

prt\_sex as sex,

/\*prt\_address1 = upper(prt\_addr1)\*/

upper(prt\_addr1) as prt\_address1,

/\*prt\_address2 = upper(prt\_addr2)\*/

upper(prt\_addr2) as prt\_address2,

prt\_middle

from hbt.provider p join hbt.demographic d

--added cast to numeric as data types are different

on kn\_number = cast(prt\_ss\_nbr as numeric)

where not exists (select 1 from pstg.person t where t.prt\_ss\_nbr = d.prt\_ss\_nbr)

);

END;

$$ LANGUAGE plpgsql;

* call pstg.Load\_AllPersonCriteria();

CREATE OR REPLACE PROCEDURE pstg.Load\_AllPersonCriteria()

AS $$

BEGIN

/\*First Pass Logic:

Matching criteria - (kn\_key\_name = substring(c.dshs\_provider\_id, 1, 7))

and records not present in person table\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,

prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

on left(t.dshs\_provider\_id, 7) = b.kn\_key\_name

and pass\_criteria=1;

update pstg.person

set contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone, mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person t

join pstg.basecontact c

on (t.kn\_key\_name = substring(c.dshs\_provider\_id, 1, 7))

and pass\_criteria=1;

insert into pstg.person(contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2,

pass\_criteria)

(SELECT DISTINCT c.contactid,

c.student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

c.birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

c.phone,

c.mobilephone,

c.homephone,

prt\_zip\_code,

substring(c.dshs\_provider\_id, 1, 7) AS dshs\_provider\_id,

cast(prt\_ss\_nbr as varchar(20)),

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2,

'1'

from pstg.basecontact c join pstg.basehbt h on (kn\_key\_name = substring(c.dshs\_provider\_id, 1, 7))

/\* For records not present in Person Table\*/

where not exists(select 1 from pstg.person b where b.kn\_key\_name = h.kn\_key\_name)

and kn\_key\_name <> '0'

and kn\_key\_name <> ''

and dshs\_provider\_id <> '0'

and dshs\_provider\_id <> ''

);

/\*

Secondpass logic:

Matching criteria- (firstname = prt\_first and lastname = prt\_last)

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

and (prt\_zip\_code = mailingpostalcode)

and (prt\_telephone\_nbr = phone or prt\_telephone\_nbr = mobilephone or prt\_telephone\_nbr = homephone) and records not present in person table

\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,

prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

on (b.prt\_first = t.firstname

and b.prt\_last = t.lastname

and b.prt\_ss\_nbr\_last4 = t.social\_security\_number\_last4

and (date(b.prt\_birth\_date) = date(t.birthdate))

and b.prt\_zip\_code = t.mailingpostalcode

and (b.prt\_telephone\_nbr = t.phone

or b.prt\_telephone\_nbr = t.mobilephone

or b.prt\_telephone\_nbr = t.homephone

)

and pass\_criteria=2

);

update pstg.person

set contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone,

mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person b

join pstg.basecontact c

on (b.prt\_first = c.firstname

and b.prt\_last = c.lastname

and b.prt\_ss\_nbr\_last4 = c.social\_security\_number\_last4

and b.prt\_birth\_date = c.birthdate

and b.prt\_zip\_code = c.mailingpostalcode

and (b.prt\_telephone\_nbr = c.phone

or b.prt\_telephone\_nbr = c.mobilephone

or b.prt\_telephone\_nbr = c.homephone)

and pass\_criteria=2

);

-- Second Criteria

insert into pstg.person(

contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2,

pass\_criteria)

(SELECT contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex, address,

prt\_address1,

prt\_address2,

'2'

FROM

(SELECT contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,

social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,

phone, mobilephone, homephone, prt\_zip\_code, dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode, prt\_first, prt\_last, zen\_age, birthdate\_age,sex, address, prt\_address1, prt\_address2, '2',

ROW\_NUMBER() OVER( PARTITION BY firstname,lastname, prt\_ss\_nbr, prt\_birth\_date, prt\_zip\_code, prt\_telephone\_nbr

ORDER BY contactid ) AS row\_num

from pstg.basehbt h join pstg.basecontact f

/\*logic as per second criteria\*/

on (firstname = prt\_first and lastname = prt\_last)

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

and (prt\_zip\_code = mailingpostalcode)

and (prt\_telephone\_nbr = phone

or prt\_telephone\_nbr = mobilephone

or prt\_telephone\_nbr = homephone)

/\*Below logics nowhere mentioned in criteria \*/

and mailingpostalcode != '0'

and mailingpostalcode != ''

and prt\_ss\_nbr <> ''

and prt\_ss\_nbr <> '0'

and prt\_telephone\_nbr <> ''

and prt\_telephone\_nbr <> '0'

and prt\_birth\_date is not null

--Get missing contacts which are not present in Person Table

and not exists (select 1 from pstg.person b where b.lastname = f.lastname and b.firstname = f.firstname and (b.prt\_ss\_nbr\_last4 = h.prt\_ss\_nbr\_last4)

and (date(b.birthdate) = date(f.birthdate))

and (b.prt\_zip\_code = h.prt\_zip\_code)

and (b.prt\_telephone\_nbr = h.prt\_telephone\_nbr))

) t

where t.row\_num < 2);

/\*

Thirdpass logic:

Matching criteria- (firstname = prt\_first and lastname = prt\_last)

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

and (prt\_zip\_code = mailingpostalcode)

and records not present in person table

\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,

prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

/\* Logic as per third pass\*/

on (b.prt\_first = t.firstname

and b.prt\_last = t.lastname

and b.prt\_ss\_nbr\_last4 = t.social\_security\_number\_last4

and b.prt\_birth\_date = t.birthdate

and b.prt\_zip\_code = t.mailingpostalcode

)

and pass\_criteria=3

;

update pstg.person

set

contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone,

mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person b

join pstg.basecontact c

/\* Logic as per third pass\*/

on (b.prt\_first = c.firstname

and b.prt\_last = c.lastname

and b.prt\_ss\_nbr\_last4 = c.social\_security\_number\_last4

and b.prt\_birth\_date = c.birthdate

and b.prt\_zip\_code = c.mailingpostalcode

)

and pass\_criteria=3

;

-- Third Criteria

insert into pstg.person(

contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,address,

prt\_address1,

prt\_address2,

pass\_criteria)

(

SELECT contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone, mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2,

'3'

FROM

(SELECT contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,

social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,

phone, mobilephone, homephone, prt\_zip\_code, dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode, prt\_first, prt\_last, zen\_age, birthdate\_age,sex, address, prt\_address1, prt\_address2, '3',

ROW\_NUMBER() OVER( PARTITION BY firstname,lastname, prt\_ss\_nbr, prt\_birth\_date, prt\_zip\_code

ORDER BY contactid ) AS row\_num

from pstg.basehbt h join pstg.basecontact f

/\* Logic as per third pass\*/

on (firstname = prt\_first and lastname = prt\_last)

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

and (prt\_zip\_code = mailingpostalcode)

/\*Logics not in thirdpass\*/

and prt\_ss\_nbr <> ''

and prt\_birth\_date is not null

and prt\_ss\_nbr <> '0'

and mailingpostalcode <> '0'

and mailingpostalcode <> ''

/\* Contacts not present in Person table \*/

and not exists(select 1 from pstg.person b where (b.firstname = f.firstname and b.lastname = f.lastname)

and (b.prt\_ss\_nbr\_last4 = h.prt\_ss\_nbr\_last4)

and (date(b.birthdate) = date(f.birthdate))

and (b.prt\_zip\_code = h.prt\_zip\_code))

)t

where t.row\_num < 2);

/\*Fourthpass logic:

Matching criteria- ((firstname = prt\_first and lastname = prt\_last)

or ((firstname+' '+mid\_name+' '+lastname) = (prt\_first+' '+prt\_mid+' '+prt\_last))

or ((firstname+' '+mid\_name+' '+lastname) = (prt\_first+' '+left(prt\_mid,1)+' '+prt\_last)))

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

and records not present in person table\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,

prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

on (b.prt\_first = t.firstname

and b.prt\_last = t.lastname

and b.prt\_ss\_nbr\_last4 = t.social\_security\_number\_last4

and b.prt\_birth\_date = t.birthdate

)

and pass\_criteria=4

;

update pstg.person

set contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone,

mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person b

join pstg.basecontact c

on (b.prt\_first = c.firstname

and b.prt\_last = c.lastname

and b.prt\_ss\_nbr\_last4 = c.social\_security\_number\_last4

and b.prt\_birth\_date = c.birthdate

)

and pass\_criteria=4

;

-- Fourth Criteria

insert into pstg.person(

contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,zen\_age,

birthdate\_age,

sex,address,

prt\_address1,

prt\_address2,

pass\_criteria)

(

SELECT

contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,address,

prt\_address1,

prt\_address2,

'4'

FROM

(SELECT contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,

social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,

phone, mobilephone, homephone, prt\_zip\_code, dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode, prt\_first, prt\_last, zen\_age, birthdate\_age,sex, address, prt\_address1, prt\_address2, '4',

ROW\_NUMBER() OVER( PARTITION BY firstname,lastname,prt\_ss\_nbr, prt\_birth\_date

ORDER BY contactid ) AS row\_num

from pstg.basehbt h join pstg.basecontact f

/\* Logic as per requirement\*/

on ((firstname = prt\_first and lastname = prt\_last)

or ((firstname+' '+mid\_name+' '+lastname) = (prt\_first+' '+prt\_mid+' '+prt\_last))

or ((firstname+' '+mid\_name+' '+lastname) = (prt\_first+' '+left(prt\_mid,1)+' '+prt\_last)))

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

/\*logic not in requirement\*/

and prt\_ss\_nbr <> ''

and prt\_birth\_date is not null

and prt\_ss\_nbr <> '0'

/\* records which are not present in person table \*/

and not exists (select 1 from pstg.person a where (f.firstname = a.firstname and f.lastname = a.lastname)

and (h.prt\_ss\_nbr\_last4 = a.prt\_ss\_nbr\_last4)

and (date(h.prt\_birth\_date) = date(a.prt\_birth\_date)))

)t

where t.row\_num < 2);

/\*

Fifthpass logic:

Matching criteria- ((lastname = prt\_last)

or ((prt\_last+' '+prt\_first) = lastname))

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

and records not present in person table

\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,

prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

on (b.prt\_last = t.lastname

and b.prt\_ss\_nbr\_last4 = t.social\_security\_number\_last4

and b.prt\_birth\_date = t.birthdate

)

and pass\_criteria=5

;

update pstg.person

set contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone,

mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person b

join pstg.basecontact c

on (b.prt\_last = c.lastname

and b.prt\_ss\_nbr\_last4 = c.social\_security\_number\_last4

and b.prt\_birth\_date = c.birthdate

)

and pass\_criteria=5

;

-- Fift Criteria

insert into pstg.person(contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,address,

prt\_address1,

prt\_address2,

pass\_criteria)

(

SELECT contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

cast(prt\_zip\_code as numeric(5)),

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2 ,

'5'

FROM

(SELECT contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,

social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,

phone, mobilephone, homephone, cast(prt\_zip\_code as numeric(5)), dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode , prt\_first, prt\_last, zen\_age, birthdate\_age, sex, address, prt\_address1, prt\_address2, '5',

ROW\_NUMBER() OVER( PARTITION BY lastname, prt\_ss\_nbr, prt\_birth\_date

ORDER BY contactid ) AS row\_num

from pstg.basehbt h join pstg.basecontact f

/\*lOGIC AS PER FIFTH CONDITION\*/

on ((lastname = prt\_last)

or ((prt\_last+' '+prt\_first) = lastname))

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

/\* logic not in requirement\*/

and prt\_ss\_nbr <> ''

and prt\_birth\_date is not null

and prt\_ss\_nbr <> '0'

/\*records not present in person table\*/

and not exists (select 1 from pstg.person a where ((f.lastname = a.lastname)

or ((a.prt\_last+' '+a.prt\_first) = (h.prt\_last+' '+h.prt\_first)))

and (a.prt\_ss\_nbr\_last4 = h.prt\_ss\_nbr\_last4)

and (date(a.birthdate) = date(f.birthdate)))

) t

where t.row\_num < 2);

/\*

Sixthpass logic:

Matching criteria- (firstname = prt\_first and lastname = prt\_last)

and (birthdate = prt\_birth\_date)

\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

on (b.prt\_first = t.firstname

and b.prt\_last = t.lastname

and b.prt\_birth\_date = t.birthdate

)

and pass\_criteria=6

;

update pstg.person

set

contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone,

mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person b

join pstg.basecontact c

on (b.prt\_first = c.firstname

and b.prt\_last = c.lastname

and b.prt\_birth\_date = c.birthdate

)

and pass\_criteria=6

;

-- Sixth Criteria

insert into pstg.person(contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,address,

prt\_address1,

prt\_address2,

pass\_criteria)

(

SELECT distinct contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2 ,

'6'

FROM

(SELECT contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,

social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,

phone, mobilephone, homephone, prt\_zip\_code, dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode, prt\_first, prt\_last, zen\_age, birthdate\_age, sex, address, prt\_address1, prt\_address2, '6',

ROW\_NUMBER() OVER( PARTITION BY firstname, lastname, prt\_birth\_date

ORDER BY contactid ) AS row\_num

FROM pstg.basehbt h join pstg.basecontact f

/\* Logic as per requirement \*/

on (firstname = prt\_first and lastname = prt\_last)

and (birthdate = prt\_birth\_date)

and prt\_birth\_date is not null

/\* Logic not in requirement \*/

and not exists (select 1 from pstg.person a where (h.prt\_first = a.prt\_first and h.prt\_last = a.prt\_last)

and (h.prt\_birth\_date = a.prt\_birth\_date))

)t

where t.row\_num < 2);

/\*

Seventhpass logic:

Matching criteria- (firstname = prt\_first and lastname = prt\_last)

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,

prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

on (b.prt\_first = t.firstname

/\* Missing Logic added \*/

and (length(b.prt\_ss\_nbr) >= 8)

and b.prt\_last = t.lastname

and b.prt\_ss\_nbr\_last4 = t.social\_security\_number\_last4

)

and pass\_criteria=7

;

update pstg.person

set contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone,

mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person b

join pstg.basecontact c

on (b.prt\_first = c.firstname

and b.prt\_last = c.lastname

/\* Missing Logic added \*/

and (length(b.prt\_ss\_nbr) >= 8)

and b.prt\_ss\_nbr\_last4 = c.social\_security\_number\_last4

)

and pass\_criteria=7

;

-- Seven Criteria

insert into pstg.person(contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2,

pass\_criteria)

(

SELECT distinct contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,phone, mobilephone, homephone, prt\_zip\_code, dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode, prt\_first, prt\_last, zen\_age, birthdate\_age, sex, address, prt\_address1, prt\_address2 , '7'

FROM

(SELECT contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,

social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,

phone, mobilephone, homephone, prt\_zip\_code, dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode, prt\_first, prt\_last, zen\_age, birthdate\_age, sex, address, prt\_address1, prt\_address2, '7',

ROW\_NUMBER() OVER( PARTITION BY firstname, lastname, prt\_ss\_nbr

ORDER BY contactid ) AS row\_num

FROM pstg.basehbt h join pstg.basecontact f

/\* Logic as per requirement \*/

on (firstname = prt\_first and lastname = prt\_last)

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

/\* Logic not in requirement \*/

and prt\_ss\_nbr <> ''

and prt\_ss\_nbr <> '0'

and not exists (select 1 from pstg.person a where (h.prt\_first = a.prt\_first and h.prt\_last = a.prt\_last)

and (h.prt\_ss\_nbr\_last4 = a.prt\_ss\_nbr\_last4))

)t

where t.row\_num < 2);

/\*

Eighthpass logic:

Matching criteria- (firstname = prt\_first)

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

and (prt\_zip\_code = mailingpostalcode)

\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,

prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

on (b.prt\_first = t.firstname

/\* Missing Logic added \*/

and (length(b.prt\_ss\_nbr) >= 8)

and b.prt\_ss\_nbr\_last4 = t.social\_security\_number\_last4

and b.prt\_birth\_date = t.birthdate

and b.prt\_zip\_code = t.mailingpostalcode

)

and pass\_criteria=8

;

update pstg.person

set contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone,

mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person b

join pstg.basecontact c

on (b.prt\_first = c.firstname

/\* Missing Logic added \*/

and (length(b.prt\_ss\_nbr) >= 8)

and b.prt\_ss\_nbr\_last4 = c.social\_security\_number\_last4

and b.prt\_birth\_date = c.birthdate

and b.prt\_zip\_code = c.mailingpostalcode)

and pass\_criteria=8

;

-- Eight Criteria

insert into pstg.person(contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date, birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,address,

prt\_address1,

prt\_address2,

pass\_criteria)

(

SELECT contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2,

'8'

FROM

(SELECT contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,

social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,

phone, mobilephone, homephone, prt\_zip\_code, dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode, prt\_first, prt\_last, zen\_age, birthdate\_age,sex, address, prt\_address1, prt\_address2, '8',

ROW\_NUMBER() OVER( PARTITION BY firstname, prt\_ss\_nbr, prt\_birth\_date, prt\_zip\_code

ORDER BY contactid ) AS row\_num

from pstg.basehbt h join pstg.basecontact f

/\* Logic as per requirement \*/

on (firstname = prt\_first)

and (length(prt\_ss\_nbr) >= 8)

and (prt\_ss\_nbr\_last4 = social\_security\_number\_last4)

and (date(birthdate) = date(prt\_birth\_date))

and (prt\_zip\_code = mailingpostalcode)

/\* Logic not in requirement \*/

and prt\_ss\_nbr <> ''

and prt\_birth\_date is not null

and prt\_ss\_nbr <> '0'

and mailingpostalcode <> '0'

and mailingpostalcode <> ''

and not exists (select 1 from pstg.person a where (h.prt\_first = a.prt\_first)

and (h.prt\_ss\_nbr\_last4 = a.prt\_ss\_nbr\_last4)

and (date(h.prt\_birth\_date) = date(a.prt\_birth\_date))

and (h.prt\_zip\_code = a.prt\_zip\_code))

) t

where t.row\_num < 2);

/\*

Ninthpass logic:

Matching criteria- (firstname = prt\_first and lastname = prt\_last)

and (prt\_zip\_code = mailingpostalcode)

and (address = prt\_address1

or address = prt\_address2

or address = (prt\_address1+' '+prt\_address2)

or address = (prt\_address2+' '+prt\_address1))

\*/

update pstg.person

set kn\_key\_name = b.kn\_key\_name,

prt\_first = b.prt\_first,

prt\_last = b.prt\_last,

prt\_ss\_nbr\_last4 = b.prt\_ss\_nbr\_last4,

prt\_birth\_date = b.prt\_birth\_date,

prt\_employer\_code = b.prt\_employer\_code,

prt\_telephone\_nbr = b.prt\_telephone\_nbr,

prt\_zip\_code = b.prt\_zip\_code,

prt\_ss\_nbr = b.prt\_ss\_nbr,

lastmodified = sysdate,

zen\_age = b.zen\_age,

sex = b.sex,

prt\_address1 = b.prt\_address1,

prt\_address2 = b.prt\_address2

from pstg.person t

join pstg.basehbt b

on (b.prt\_first = t.firstname

and b.prt\_last = t.lastname

and b.prt\_zip\_code = t.mailingpostalcode

and (b.prt\_address1 = t.address

or b.prt\_address2 = t.address

or (b.prt\_address1+' '+b.prt\_address2) = t.address

or (b.prt\_address2+' '+b.prt\_address1) = t.address

)

and pass\_criteria=9

);

update pstg.person

set contactid = c.contactid,

student\_id = c.student\_id,

firstname = c.firstname,

lastname = c.lastname,

social\_security\_number\_last4 = c.social\_security\_number\_last4,

birthdate = c.birthdate,

phone = c.phone,

mobilephone = c.mobilephone,

homephone = c.homephone,

dshs\_provider\_id = c.dshs\_provider\_id,

mailingpostalcode = c.mailingpostalcode,

lastmodified = sysdate,

birthdate\_age = c.birthdate\_age,

address = c.address

from pstg.person b

join pstg.basecontact c

on (b.prt\_first = c.firstname

and b.prt\_last = c.lastname

and (b.prt\_address1 = c.address

or b.prt\_address2 = c.address

or (b.prt\_address1+' '+b.prt\_address2) = c.address

or (b.prt\_address2+' '+b.prt\_address1) = c.address

)

and pass\_criteria=9

);

-- Ninth Criteria

insert into pstg.person(contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date, birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,

phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2,

pass\_criteria)

(

SELECT contactid,

student\_id,

kn\_key\_name,

firstname,

lastname,

prt\_ss\_nbr\_last4,

social\_security\_number\_last4,

prt\_birth\_date,

birthdate,

prt\_employer\_code,

prt\_telephone\_nbr,phone,

mobilephone,

homephone,

prt\_zip\_code,

dshs\_provider\_id,

prt\_ss\_nbr,

mailingpostalcode,

prt\_first,

prt\_last,

zen\_age,

birthdate\_age,

sex,

address,

prt\_address1,

prt\_address2,

'9'

FROM

(SELECT contactid, student\_id, kn\_key\_name, firstname, lastname, prt\_ss\_nbr\_last4,

social\_security\_number\_last4, prt\_birth\_date, birthdate, prt\_employer\_code, prt\_telephone\_nbr,

phone, mobilephone, homephone, prt\_zip\_code, dshs\_provider\_id, prt\_ss\_nbr, mailingpostalcode, prt\_first, prt\_last, zen\_age, birthdate\_age,sex, address, prt\_address1, prt\_address2, '9',

ROW\_NUMBER() OVER( PARTITION BY firstname,lastname, prt\_zip\_code, address

ORDER BY contactid ) AS row\_num

from pstg.basehbt h join pstg.basecontact f

/\*Logic as per requirement \*/

on (firstname = prt\_first and lastname = prt\_last)

and (prt\_zip\_code = mailingpostalcode)

and (address = prt\_address1

or address = prt\_address2

or address = (prt\_address1+' '+prt\_address2)

or address = (prt\_address2+' '+prt\_address1)

)

/\*Logic not in requirmeent \*/

and address <> ''

and address <> '0'

and mailingpostalcode <> '0'

and mailingpostalcode <> ''

and not exists (select 1 from pstg.person a where (h.prt\_first = a.prt\_first and h.prt\_last = a.prt\_last)

and (h.prt\_zip\_code = a.prt\_zip\_code)

and (h.prt\_address1 = a.prt\_address1

or h.prt\_address2 = a.prt\_address2

or (h.prt\_address1+' '+h.prt\_address2) = (a.prt\_address1+' '+a.prt\_address2)

or (h.prt\_address2+' '+h.prt\_address1) = (a.prt\_address2+' '+a.prt\_address1)

))

)t

where t.row\_num < 2);

/\*

HBT nonmatched records logic:

Matching criteria- (select \* from pstg.basehbt b

where not exists(select 1 from pstg.person t where b.prt\_ss\_nbr = t.prt\_ss\_nbr and b.kn\_key\_name = t.kn\_key\_name)

and prt\_first <> ''

and prt\_last <> '')\*/

--HBT nonmatched records

insert into pstg.person(kn\_key\_name,

prt\_ss\_nbr\_last4,

prt\_birth\_date,

prt\_employer\_code,

prt\_telephone\_nbr,

prt\_zip\_code,

prt\_ss\_nbr,

prt\_first,

prt\_last,

zen\_age,

sex,

prt\_address1,

prt\_address2,

pass\_criteria)

(select kn\_key\_name, right(prt\_ss\_nbr, 4) as prt\_ss\_nbr\_last4, prt\_birth\_date, prt\_employer\_code, prt\_telephone\_nbr, prt\_zip\_code, prt\_ss\_nbr, prt\_first, prt\_last,zen\_age, sex, prt\_address1, prt\_address2, 'NULL'

from pstg.basehbt b

where not exists(select 1 from pstg.person t where b.prt\_ss\_nbr = t.prt\_ss\_nbr and b.kn\_key\_name = t.kn\_key\_name)

and prt\_first <> ''

and prt\_last <> '');

/\*

Contact nonmatched records logic:

Matching criteria- (select \* from pstg.basecontact f

where not exists (select 1 from pstg.person t where f.student\_id = t.student\_id))

\*/

--Contact nonmatched records logic:

insert into pstg.person(contactid,

student\_id,

firstname,

lastname,

social\_security\_number\_last4,

birthdate,

phone,

mobilephone,

homephone,

dshs\_provider\_id,

mailingpostalcode,

birthdate\_age,

address,

pass\_criteria)

(

select contactid, student\_id, firstname, lastname,

social\_security\_number\_last4, birthdate,

phone, mobilephone, homephone, dshs\_provider\_id, mailingpostalcode, birthdate\_age, address, 'NULL'

/\* Changed the logic from contact id to student id as per requirements \*/

from pstg.basecontact f where not exists (select 1 from pstg.person t where f.student\_id= t.student\_id)

)

;

update pstg.person set pass\_criteria = split\_part(pass\_criteria, '.', 1) from pstg.person where pass\_criteria LIKE '%.%';

update pstg.person

set pass\_criteria = (case when n.pass\_criteria = 'NULL' then '0'

when n.pass\_criteria =0 then '0'

else n.pass\_criteria ||'.'||row\_id end)

from

(select \*, row\_number() over (partition by contactid) row\_id

from

(select \*,

count(contactid) over (partition by contactid, pass\_criteria) rn

from pstg.person

where contactid in (Select contactid from pstg.person pk group by pk.contactid having count(pk.contactid) > 1))

where rn > 1) n

join pstg.person p on p.prt\_ss\_nbr = n.prt\_ss\_nbr;

update pstg.person

set pass\_criteria = (case when n.pass\_criteria = 'NULL' then '0'

when n.pass\_criteria =0 then '0'

else n.pass\_criteria ||'.'||row\_id end)

from

(select \*, row\_number() over (partition by kn\_key\_name) row\_id

from

(select \*,

count(kn\_key\_name) over (partition by kn\_key\_name, pass\_criteria) rn

from pstg.person

where kn\_key\_name in (Select kn\_key\_name from pstg.person pk group by pk.kn\_key\_name having count(pk.kn\_key\_name) > 1))

where rn > 1) n

join pstg.person p on p.contactid = n.contactid;

/\*First logic : Update Personid for contactid duplicates set personid= m.bgidkey\*/

update pstg.person

set personid= m.bgidkey

from

(select \* from

(select \*, row\_number() over (partition by contactid order by pass\_criteria) as rn

from pstg.person

/\*Find duplicate for contactid \*/

where contactid in (select contactid from pstg.person group by contactid having count(contactid ) > 1)) t

where t.rn = 1) m

join pstg.person x on x.contactid = m.contactid

and x.bgidkey != m.bgidkey

where m.contactid in

(select contactid from

(select \*, row\_number() over (partition by contactid order by pass\_criteria) as rn

from pstg.person

where contactid in (select contactid from pstg.person group by contactid having count(contactid ) > 1)) t

where t.rn > 1

);

/\*Second logic : Update Personid for kn\_key\_name duplicates set personid= m.bgidkey\*/

update pstg.person

set personid= m.bgidkey

from

(select \* from

(select \*, row\_number() over (partition by kn\_key\_name order by pass\_criteria) as rn

from pstg.person

/\*Find duplicate for fn\_key\_name \*/

where kn\_key\_name in (select kn\_key\_name from pstg.person group by kn\_key\_name having count(kn\_key\_name ) > 1)) t

where t.rn = 1) m

join pstg.person x on x.kn\_key\_name = m.kn\_key\_name

and x.bgidkey != m.bgidkey

where m.kn\_key\_name in

(select kn\_key\_name from

(select \*, row\_number() over (partition by kn\_key\_name order by pass\_criteria) as rn

from pstg.person

where kn\_key\_name in (select kn\_key\_name from pstg.person group by kn\_key\_name having count(kn\_key\_name ) > 1)) t

where t.rn > 1

);

/\* Third Logic : Update Personid with the default bgidkey for non duplicates set personid= m.bgidkey\*/

update pstg.person set personid = bgidkey where personid is null;

/\* Fourth Logic : Update keyindicator for duplicates set keyindicator = 'true'\*/

update pstg.person

set keyindicator= 'true'

from pstg.person t

where kn\_key\_name in

/\* Check for duplicates with contact id >1 or by kn\_key\_name>1\*/

(select kn\_key\_name from pstg.person group by kn\_key\_name having count(kn\_key\_name) > 1)

or contactid in

(select contactid from pstg.person group by contactid having count(contactid) > 1);

/\* Fifth Logic : Update keyindicator for non duplicates set keyindicator = 'false'\*/

update pstg.person set keyindicator = 'false' from pstg.person t where keyindicator is null;

END;

$$ LANGUAGE plpgsql;