

Generate a custom Organisation chart from ps_employees

In the PS_EMPLOYEES table, there is a custom field called SUPERVISOR_ID. This is used to connect the employee to the supervisor or line manager. This is similar to the REPORTS_TO field in PS_JOB.

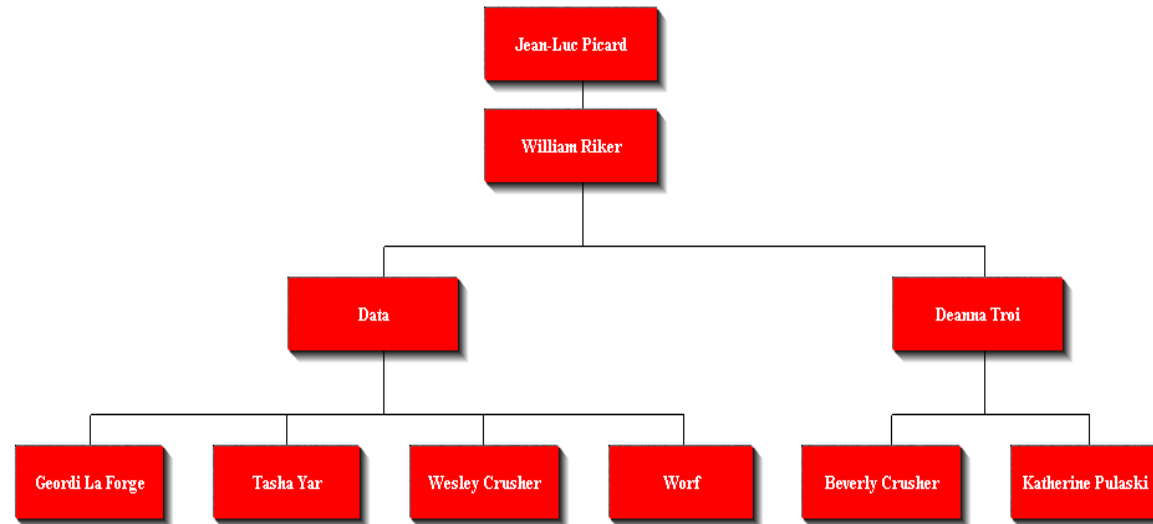
I developed this after toying with the GOOGLE API Organisation Chart. Generating the boxes was straight forward enough, so the trick was to understand how to generate the connecting lines.

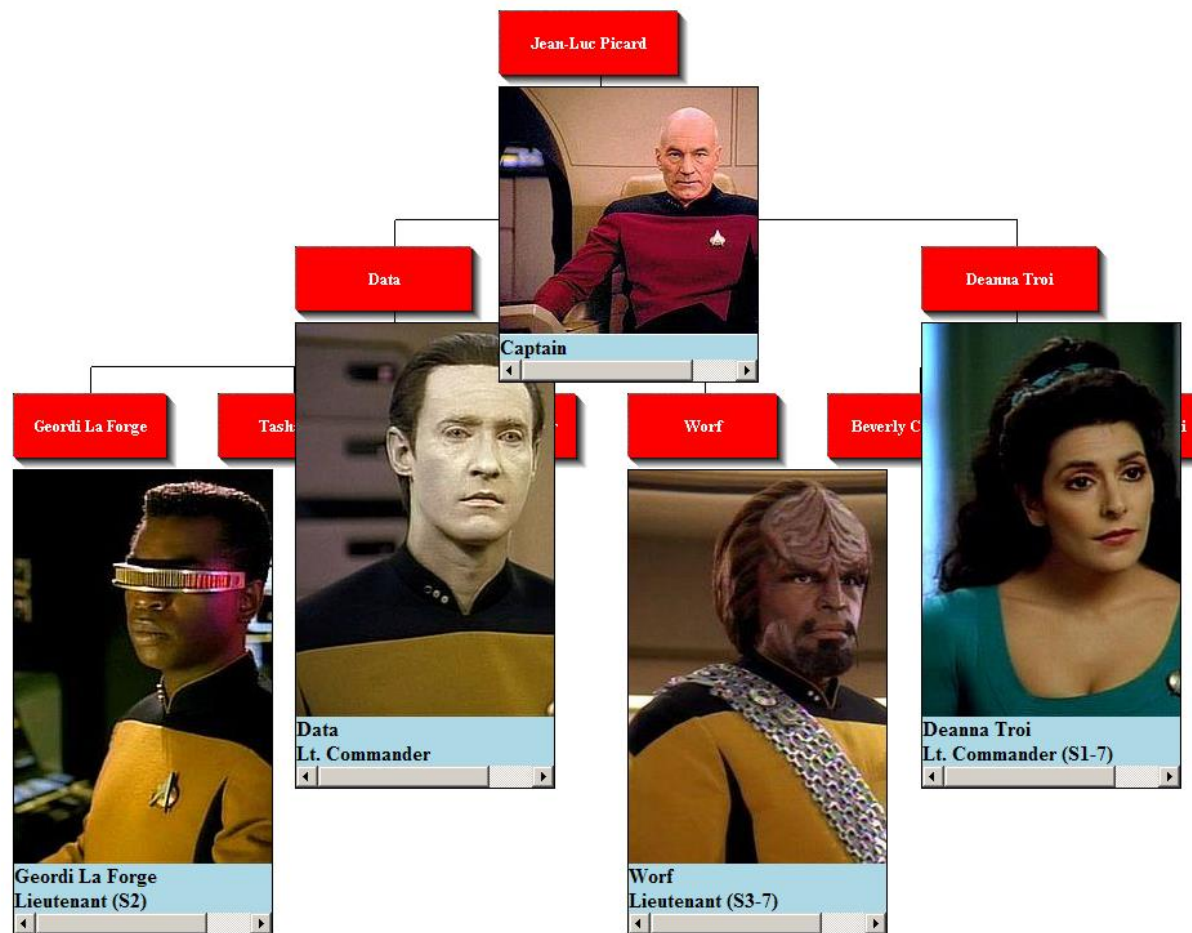
I also included some Javascript to make the images appear when the box is clicked and disappear when it is double-clicked. The images are from the PS_EMPL_PHOTO, so a data URI with Base64 encoding is used.

Complete listing of codes are included at the bottom of this document. Once you saved the listing, you run it in SQL*PLUS after you have set up the 3 tables I use, and changing all reference to the hard-coded EMPLID in the example '11528978'. Also Step 22 requires 3 PL/SQL to generate images from ps_empl_photo, so you can comment out.

Line Manager Connection

Click on box to show details
Double-click on box to hide details





Explanation of key steps (for a specific employee)

Note: I have a SQR job based on this to connect ~ 30000 people which runs in ~10 mins using job streaming.

Populating PS_RY_TREE_WRK0 – Step 1 in listing

A Global Temporary Table, PS_RY_TREE_WRK0 is used as a base table, to store the output from the following sql. The connection is from the employee to his/her line manager using the supervisor_id field via the “connect by nocycle” statement and restricting the connection level to 3.

```
truncate table ps_ry_tree_wrk0
;
insert /*populate wrk0*/ into ps_ry_tree_wrk0
select rownum , mylevel, id, id_name, parent_id, parent_name
from
(
select
    level mylevel
, a.emplid          id
, a.name            id_name
, a.supervisor_id   parent_id
, coalesce((select b.name from ps_employees b where b.emplid = a.supervisor_id),' ') parent_name
, connect_by_iscycle
from ps_employees a
where a.empl_status = 'A'
start with a.emplid = '11528978'
connect by nocycle prior a.emplid = a.supervisor_id and level <= 3
order by level
)
;
commit
;
```

Now, the information in ps_ry_tree_wrk0 looks like

```
SQL> select * from ps_ry_tree_wrk0
2 /
```

MYROWNUM	MYLEVEL	ID	ID_NAME	PARENT_ID	PARENT_NAME
1	1	11528978	William Riker	11530058	Jean-Luc Picard
2	2	1153776A	Deanna Troi	11528978	William Riker
3	2	11531130	Data	11528978	William Riker
4	3	11531405	Worf	11531130	Data
5	3	11536310	Katherine Pulaski	1153776A	Deanna Troi
6	3	11537786	Beverly Crusher	1153776A	Deanna Troi
7	3	11531149	Wesley Crusher	11531130	Data
8	3	11529065	Geordi La Forge	11531130	Data
9	3	11531181	Tasha Yar	11531130	Data

9 rows selected.

I now used a second table, PS_RY_TREE_WRK, to extract data from the Step 1, and in addition, store additional details like children/grandchildren counts, first/last child indicators, maximum widths for each fow.

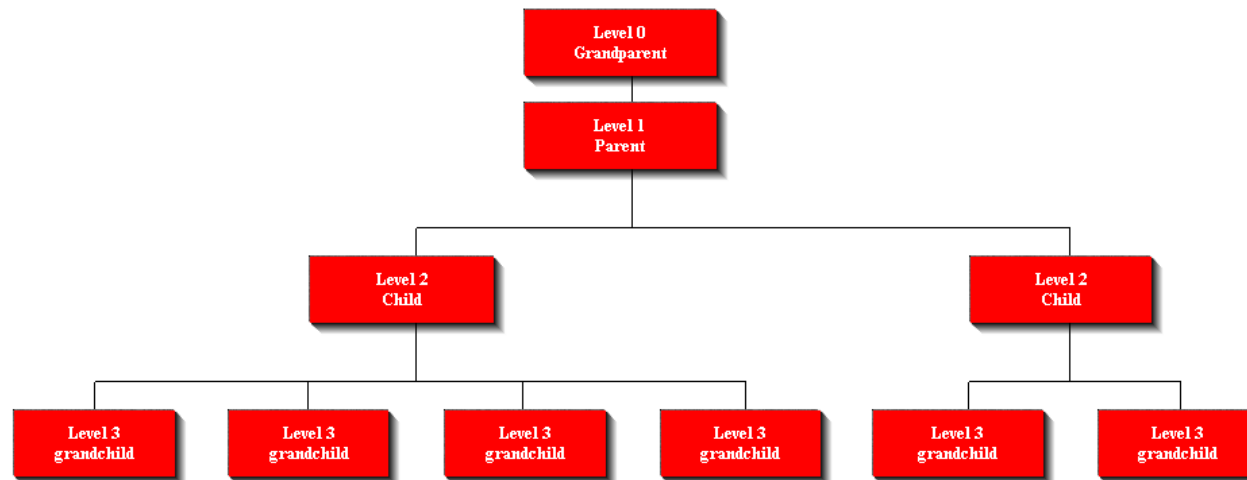
```
SQL> descr ps_ry_tree_wrk;
```

Name	Null?	Type
MYLEVEL	NOT NULL	NUMBER(38)
MYROWNUM	NOT NULL	NUMBER(38)
ID	NOT NULL	VARCHAR2(11 CHAR)
ID_NAME	NOT NULL	VARCHAR2(50 CHAR)
SEQ	NOT NULL	NUMBER(38)
PARENT_ID	NOT NULL	VARCHAR2(11 CHAR)
PARENT_NAME	NOT NULL	VARCHAR2(50 CHAR)
PARENT_SEQ	NOT NULL	NUMBER(38)
CHILD_CNT	NOT NULL	NUMBER(38)
G_CHILD_CNT	NOT NULL	NUMBER(38)
GG_CHILD_CNT	NOT NULL	NUMBER(38)
MAX_CNT	NOT NULL	NUMBER(38)
FIRST_CHILD	NOT NULL	VARCHAR2(1 CHAR)
LAST_CHILD	NOT NULL	VARCHAR2(1 CHAR)
MID_POINT	NOT NULL	NUMBER(38)

Line Manager Connection

Click on box to show details

Double-click on box to hide details



Populate ps_ry_tree_wrk from ps_ry_tree_wrk0 for Level 1 only – Step 2 in listing

```
truncate table ps_ry_tree_wrk
```

```
;
```

```
insert /*populate wrk level 1*/ into ps_ry_tree_wrk
```

```
select
```

```
mylevel, rownum, id, id_name , rownum seq, parent_id, parent_name, 1 parent_seq, 0, 0, 0, 0, '', '', 0
```

```
from
```

```
(
```

```
select
```

```
a.myrownum
```

```

, a.mylevel
, a.id
, a.id_name
, a.parent_id
, a.parent_name
from ps_ry_tree_wrk0 a
where a.id = '11528978'
and a.mylevel = 1
)

```

Populate ps_ry_tree_wrk for Level 2 only using Level 1 data – Step 3 in listing

```

insert /*populate wrk level 1*/ into ps_ry_tree_wrk
select
mylevel, rownum, id, id_name , seq, parent_id, parent_name, parent_seq, 0, 0, 0, 0, ' ', ' ', 0
from
(
select
a.mylevel +1      mylevel
, case when b.id    is null then '00000000' else b.id      end    id
, case when b.id_name  is null then 'xxxxxxxx' else b.id_name  end    id_name
, rank() over (partition by a.mylevel +1, a.myrownum, a.parent_seq, a.seq order by a.mylevel +1, a.myrownum, a.parent_seq, a.seq, b.id_name )
seq
, a.id            parent_id
, a.id_name       parent_name
, a.myrownum      parent_seq
from ps_ry_tree_wrk a, ps_ry_tree_wrk0 b
where a.mylevel    = 1
and b.mylevel(+)   = a.mylevel +1
and b.parent_id(+) = a.id
)

```

Populate ps_ry_tree_wrk for level 3 only using level 2 data – Step 4 in listing

```
insert /*populate wrk level 1*/ into ps_ry_tree_wrk
select
mylevel, rownum, id, id_name , seq, parent_id, parent_name, parent_seq, 0, 0, 0, 0, ' ', ' ', 0
from
(
select
a.mylevel +1          mylevel
, case when b.id       is null then '00000000' else b.id       end    id
, case when b.id_name  is null then 'xxxxxxx' else b.id_name  end    id_name
, rank() over (partition by a.mylevel +1, a.myrownum, a.parent_seq, a.seq order by a.mylevel +1, a.myrownum, a.parent_seq, a.seq, b.id_name )
seq
, a.id                parent_id
, a.id_name           parent_name
, a.myrownum          parent_seq
from ps_ry_tree_wrk a, ps_ry_tree_wrk0 b
where a.mylevel       = 2
and b.mylevel(+)      = a.mylevel +1
and b.parent_id(+)    = a.id
)
```

Update first and last child information for ps_ry_tree_wrk – Step 5 in listing

1st child is when seq equals 1. Last child is located using max(seq) function.

```
update /* first child */ ps_ry_tree_wrk a
set a.first_child = case when a.seq = 1 then 'Y' else 'N' end
;
update /* last child */ ps_ry_tree_wrk a
```

set last_child = case when a.seq = coalesce((select **max(seq)** from ps_ry_tree_wrk b where b.parent_id = a.parent_id and b.mylevel = a.mylevel),1) then 'Y' else 'N' end

Update child, grand child and great grandchild count for ps_ry_tree_wrk – Step 6 in listing

```
update /* child count */ ps_ry_tree_wrk a
set child_cnt = coalesce((select count(*) from ps_ry_tree_wrk b
                        where b.mylevel = a.mylevel +1 and b.parent_id = a.id and b.parent_seq = a.myrownum),0)
;
update /* great child count */ ps_ry_tree_wrk a
set g_child_cnt = coalesce((select count(*) from ps_ry_tree_wrk b, ps_ry_tree_wrk c
                        where b.mylevel = a.mylevel +1 and b.parent_id = a.id and b.parent_seq = a.myrownum
                        and c.mylevel = b.mylevel +1 and c.parent_id = b.id and c.parent_seq = b.myrownum),0)
;
update /* great grandchild */ ps_ry_tree_wrk a
set gg_child_cnt = coalesce((select count(*) from ps_ry_tree_wrk b, ps_ry_tree_wrk c, ps_ry_tree_wrk d
                        where b.mylevel = a.mylevel +1 and b.parent_id = a.id and b.parent_seq = a.myrownum
                        and c.mylevel = b.mylevel +1 and c.parent_id = b.id and c.parent_seq = b.myrownum
                        and d.mylevel = c.mylevel +1 and d.parent_id = c.id and d.parent_seq = c.myrownum),0)
;
```

Work out maximum width based on child, grandchild and great grandchild – Step 7 in listing

```
update /* get maximum width */ ps_ry_tree_wrk a
set max_cnt = (select case when b.myrownum = 1 then
                        case when greatest(b.child_cnt, b.g_child_cnt, b.gg_child_cnt) = 0 then 8
                        else greatest(b.child_cnt, b.g_child_cnt, b.gg_child_cnt) * 8
                        end
                else
                case when greatest(b.child_cnt, b.g_child_cnt, b.gg_child_cnt) = 0 then 8
                else greatest(b.child_cnt, b.g_child_cnt, b.gg_child_cnt) *8
                end
                )
```

```

end
from ps_ry_tree_wrk b
where b.mylevel = a.mylevel
and b.myrownum = a.myrownum
and b.id = a.id
)

```

MYLEVEL	MYROWNUM	ID	SUBSTR(ID_NAME,1,25)	SEQ	PARENT_ID	SUBSTR(PARENT_NAME,1,25)	PARENT_SEQ	CHILD_CNT	G_CHILD_CNT	GG_CHILD_CNT	MAX_CNT	F	L	MID_POINT
1	1	11528978	William Riker	1	11530058	Jean-Luc Picard	1	2	6	0	48	Y	Y	23
2	1	11531130	Data	1	11528978	William Riker	1	4	0	0	32	Y	N	15
2	2	1153776A	Deanna Troi	2	11528978	William Riker	1	2	0	0	16	N	Y	7
3	1	11529065	Geordi La Forge	1	11531130	Data	1	0	0	0	8	Y	N	3
3	2	11531181	Tasha Yar	2	11531130	Data	1	0	0	0	8	N	N	3
3	3	11531149	Wesley Crusher	3	11531130	Data	1	0	0	0	8	N	N	3
3	4	11531405	Worf	4	11531130	Data	1	0	0	0	8	N	Y	3
3	5	11537786	Beverly Crusher	1	1153776A	Deanna Troi	2	0	0	0	8	Y	N	3
3	6	11536310	Katherine Pulaski	2	1153776A	Deanna Troi	2	0	0	0	8	N	Y	3

At leve 1, the ID is 11528978. It has 2 children and 6 grandchildren. The maximum width or max_cnt need for this box is $6*8 = 48$

At level 2

So, ID 11531130 (Data) has a SEQ of 1 (i.e. 1st child, first_cnt = 'Y'), with a parent of 11528978 (William Riker). Data has 4 children (child_cnt = 4), so the max_cnt is $4*8 = 32$.

ID 1153776A(Troi) has a SEQ of 2 (i.e. 2nd child and also last child, so first_cnt = 'N' and last_child_cnt = 'Y') of parent 11528978 (William Riker). Troi has 2 children (child_cnt = 2), so the max_cnt = $2*8 = 16$.

All the level 3 IDs have no children.

Here, I used my 3rd and final table to store the generated HTML for the employee. This table is structured in such way so that it can be as flexible as possible using 2 fields in particular, seq and seq2.

```
SQL> descr ps_ry_lnmgr_tree;
```

Name	Null?	Type
-----	-----	-----
EMPLID		NOT NULL VARCHAR2(11 CHAR)
SEQ		NOT NULL NUMBER(38)
SEQ2		NOT NULL NUMBER(38)
TREE_HTML		NOT NULL CLOB

Boxes and connectors below are simply cells linked to CSS (see Step 8 in listing)

Generate the Boxes – Steps 14, 17 & 20

Consider Level 3 (the last level). On this level, there are no children. So each box consists of 6 cells (i.e. <td colspan="6">) followed by another 2 blank cells if it has a valid Id, otherwise we simply generate 8 cells using colspan="8".

Level 1 & 2 have children/grandchildren, so the maximum width needs to be take into consideration generating the boxes.

Generate horizontal connectors – Steps 15 & 18

This is the most complex piece. It uses the first and last child in the current level, the mid-points of the current level and the level above and offsetting appropriately to locate the starting and ending positions of the current level (the '-' connector) and mid-point above.

Generate vertical connectors - Steps 16 & 19

This is similar to 'Generate the Boxes'.

The onclick and ondblclick javascript scripts are in Step 8.

The images from ps_empl_photo are generated as in-line using the following PL/SQL – Step 22

```
CREATE OR REPLACE FUNCTION base64encode(p_blob IN BLOB)
  RETURN CLOB
IS
  l_clob CLOB;
  l_step PLS_INTEGER := 12000;
BEGIN
  FOR i IN 0 .. TRUNC((DBMS_LOB.getlength(p_blob) - 1 )/l_step) LOOP
    l_clob := l_clob || UTL_RAW.cast_to_varchar2(UTL_ENCODE.base64_encode(DBMS_LOB.substr(p_blob, l_step, i * l_step + 1)));
  END LOOP;
  RETURN l_clob;
END;
/
```

```
CREATE OR REPLACE PROCEDURE get_enc_img_from_tab (p_image_name IN VARCHAR2,
                                                    p_clob      IN OUT NOCOPY CLOB)
AS
BEGIN
  SELECT p_clob || base64encode(employee_photo)
  INTO   p_clob
  FROM   ps_empl_photo
  WHERE  emplid = p_image_name;
END;
/
```

```
CREATE OR REPLACE FUNCTION get_ps_empl_photo(in_emplid VARCHAR2)
  RETURN CLOB
```

```

IS l_clob CLOB;
BEGIN
  DBMS_LOB.createtemporary(l_clob, FALSE);

  -- Build the start of the HTML document, including the start of the IMG tag
  -- and place it in a CLOB.
  l_clob := '';
  RETURN l_clob;

EXCEPTION
when NO_DATA_FOUND then
  return null; -- if you want to return null on nothing found
  WHEN OTHERS THEN
    DBMS_LOB.freetemporary(l_clob);
  -- RAISE;
  return null;
END;
/

```

Complete listing

```

set feedback off echo off define $ wrap on linesize 1000 pagesize 0 trimspool on LONGCHUNKSIZE 1000000 long 1000000
--Step by step guide
--

```

```
-- You can simply run this in SQL PLUS using the @command e.g. @c:\generate_lnmgr_connection.sql (whatever you have saved this as),
-- bearing in mind, a) I have used a specific employee id
-- b) I have referenced the following Peoplesoft tables
-- ps_employees, ps_empl_photo, ps_ry_tree_wrk0, ps_ry_tree_wrk and ps_ry_lnmgr_tree
--
--
--
CREATE GLOBAL TEMPORARY TABLE "SYSADM"."PS_RY_TREE_WRK0"
( "MYROWNUM" NUMBER(*,0) NOT NULL ENABLE,
  "MYLEVEL" NUMBER(*,0) NOT NULL ENABLE,
  "ID" VARCHAR2(11 CHAR) NOT NULL ENABLE,
  "ID_NAMENAME" VARCHAR2(50 CHAR) NOT NULL ENABLE,
  "PARENT_ID" VARCHAR2(11 CHAR) NOT NULL ENABLE,
  "PARENT_NAME" VARCHAR2(50 CHAR) NOT NULL ENABLE
) ON COMMIT PRESERVE ROWS

--
--
CREATE GLOBAL TEMPORARY TABLE "SYSADM"."PS_RY_TREE_WRK"
( "MYLEVEL" NUMBER(*,0) NOT NULL ENABLE,
  "MYROWNUM" NUMBER(*,0) NOT NULL ENABLE,
  "ID" VARCHAR2(11 CHAR) NOT NULL ENABLE,
  "ID_NAME" VARCHAR2(50 CHAR) NOT NULL ENABLE,
  "SEQ" NUMBER(*,0) NOT NULL ENABLE,
  "PARENT_ID" VARCHAR2(11 CHAR) NOT NULL ENABLE,
  "PARENT_NAME" VARCHAR2(50 CHAR) NOT NULL ENABLE,
  "PARENT_SEQ" NUMBER(*,0) NOT NULL ENABLE,
  "CHILD_CNT" NUMBER(*,0) NOT NULL ENABLE,
  "G_CHILD_CNT" NUMBER(*,0) NOT NULL ENABLE,
  "GG_CHILD_CNT" NUMBER(*,0) NOT NULL ENABLE,
  "MAX_CNT" NUMBER(*,0) NOT NULL ENABLE,
  "FIRST_CHILD" VARCHAR2(1 CHAR) NOT NULL ENABLE,
```

```

--      "LAST_CHILD" VARCHAR2(1 CHAR) NOT NULL ENABLE,
--      "MID_POINT" NUMBER(*,0) NOT NULL ENABLE
--  ) ON COMMIT PRESERVE ROWS
--
--  CREATE TABLE "SYSADM"."PS_RY_LNMGR_TREE"
--  (   "EMPLID" VARCHAR2(11 CHAR) NOT NULL ENABLE,
--      "SEQ" NUMBER(*,0) NOT NULL ENABLE,
--      "SEQ2" NUMBER(*,0) NOT NULL ENABLE,
--      "TREE_HTML" CLOB NOT NULL ENABLE
--  ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
--  STORAGE(INITIAL 40960 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
--  PCTINCREASE 0 BUFFER_POOL DEFAULT)
--  TABLESPACE "HRLARGE"
--  LOB ("TREE_HTML") STORE AS (
--  ENABLE STORAGE IN ROW CHUNK 8192 PCTVERSION 10
--  NOCACHE LOGGING
--  STORAGE(
--  BUFFER_POOL DEFAULT))
--  note: PS_RY_LNMGR_TREE is a partitioned table so DDL listed above is incomplete. It could have been
--      a normal table except this is part of a streamed batch job (12 streams) - better performance
--      together with the global temporary tables above
--
-- Step by step explanation
-----
-- 1. Populate ps_ry_tree_wrk0
--   Connection : emplid and supervisor_id in ps_employees
--   Start connection with specified emplid (in this example below )
--   Max connection level = 3 (in this example below )
-----
truncate table ps_ry_tree_wrk0

```

```

;
insert /*populate wrk0*/ into ps_ry_tree_wrk0
select rownum , mylevel, id, id_name, parent_id, parent_name
from
(
select
    level mylevel
  , a.emplid          id
  , a.name            id_name
  , a.supervisor_id   parent_id
  , coalesce((select b.name from ps_employees b where b.emplid = a.supervisor_id),'') parent_name
  , connect_by_iscycle
from ps_employees a
where a.empl_status = 'A'
start with a.emplid = '11528978'
connect by nocycle prior a.emplid = a.supervisor_id and level <= 3
order by level
)
;
commit
;
-----
-- 2. Populate ps_ry_tree_wrk from ps_ry_tree_wrk0 for Level 1 only
-----
truncate table ps_ry_tree_wrk
;
insert /*populate wrk level 1*/ into ps_ry_tree_wrk
select
mylevel, rownum, id, id_name , rownum seq, parent_id, parent_name, 1 parent_seq, 0, 0, 0, 0, '', '', 0
from

```

```
(
select
a.myrownum
, a.mylevel
, a.id
, a.id_name
, a.parent_id
, a.parent_name
from ps_ry_tree_wrk0 a
where a.id = '11528978'
and a.mylevel = 1
)
;
commit
;
```

```
-----
-- 3. Populate ps_ry_tree_wrk for Level 2 only using Level 1 data
-----
```

```
insert /*populate wrk level 1*/ into ps_ry_tree_wrk
select
mylevel, rownum, id, id_name , seq, parent_id, parent_name, parent_seq, 0, 0, 0, 0, '', '', 0
from
(
select
a.mylevel +1      mylevel
, case when b.id      is null then '00000000' else b.id      end      id
, case when b.id_name  is null then 'xxxxxxx' else b.id_name  end      id_name
, rank() over (partition by a.mylevel +1, a.myrownum, a.parent_seq, a.seq order by a.mylevel +1, a.myrownum, a.parent_seq, a.seq, b.id_name )
seq
, a.id            parent_id
```

```

, a.id_name      parent_name
, a.myrownum     parent_seq
from ps_ry_tree_wrk a, ps_ry_tree_wrk0 b
where a.mylevel  = 1
   and b.mylevel(+) = a.mylevel +1
   and b.parent_id(+) = a.id
)
;
commit
;

```

```

-----
-- 4. Populate ps_ry_tree_wrk for level 3 only using level 2 data
-----

```

```

insert /*populate wrk level 1*/ into ps_ry_tree_wrk
select
mylevel, rownum, id, id_name , seq, parent_id, parent_name, parent_seq, 0, 0, 0, 0, '', '', 0
from
(
select
a.mylevel +1      mylevel
, case when b.id      is null then '00000000' else b.id      end      id
, case when b.id_name  is null then 'xxxxxxx' else b.id_name  end      id_name
, rank() over (partition by a.mylevel +1, a.myrownum, a.parent_seq, a.seq order by a.mylevel +1, a.myrownum, a.parent_seq, a.seq, b.id_name )
seq
, a.id            parent_id
, a.id_name       parent_name
, a.myrownum      parent_seq
from ps_ry_tree_wrk a, ps_ry_tree_wrk0 b
where a.mylevel    = 2
   and b.mylevel(+) = a.mylevel +1

```

```

    and b.parent_id(+) = a.id
)
;
commit
;
-----
-- 5. Update first and last child information for ps_ry_tree_wrk
-----
update /* first child */ ps_ry_tree_wrk a
set a.first_child = case when a.seq = 1 then 'Y' else 'N' end
;
update /* last child */ ps_ry_tree_wrk a
set last_child = case when a.seq = coalesce((select max(seq) from ps_ry_tree_wrk b where b.parent_id = a.parent_id and b.mylevel =
a.mylevel),1) then 'Y' else 'N' end
;
commit
;
-----
-- 6. Update child, grand child and great grandchild count for ps_ry_tree_wrk
-----
update /* child count */ ps_ry_tree_wrk a
set child_cnt = coalesce((select count(*) from ps_ry_tree_wrk b
                        where b.mylevel = a.mylevel +1 and b.parent_id = a.id and b.parent_seq = a.myrownum),0)
;
update /* great child count */ ps_ry_tree_wrk a
set g_child_cnt = coalesce((select count(*) from ps_ry_tree_wrk b, ps_ry_tree_wrk c
                        where b.mylevel = a.mylevel +1 and b.parent_id = a.id and b.parent_seq = a.myrownum
                        and c.mylevel = b.mylevel +1 and c.parent_id = b.id and c.parent_seq = b.myrownum),0)
;
update /* great grandchild */ ps_ry_tree_wrk a

```

```

set gg_child_cnt = coalesce((select count(*) from ps_ry_tree_wrk b, ps_ry_tree_wrk c, ps_ry_tree_wrk d
                             where b.mylevel = a.mylevel +1 and b.parent_id = a.id and b.parent_seq = a.myrownum
                             and c.mylevel = b.mylevel +1 and c.parent_id = b.id and c.parent_seq = b.myrownum
                             and d.mylevel = c.mylevel +1 and d.parent_id = c.id and d.parent_seq = c.myrownum),0)
;
-----
-- 7. Work out maximum width based on child, grandchild and great grandchild
-----
update /* get maximum width */ ps_ry_tree_wrk a
set max_cnt = (select case when b.myrownum = 1 then
                        case when greatest(b.child_cnt, b.g_child_cnt, b.gg_child_cnt) = 0 then 8
                             else greatest(b.child_cnt, b.g_child_cnt, b.gg_child_cnt) * 8
                        end
                    else
                        case when greatest(b.child_cnt, b.g_child_cnt, b.gg_child_cnt) = 0 then 8
                             else greatest(b.child_cnt, b.g_child_cnt, b.gg_child_cnt) *8
                        end
                    end
                from ps_ry_tree_wrk b
                where b.mylevel = a.mylevel
                   and b.myrownum = a.myrownum
                   and b.id      = a.id
                )
;
commit
;
update /* set mid point */ ps_ry_tree_wrk a
set a.mid_point = ((a.max_cnt - 6) / 2) +2
;
commit

```

```

;
-----
-- 8. Generate HTML (including CSS i.e. <style>
--   Store in ps_ry_lnmgr_tree
-----
delete /* old information */ from ps_ry_lnmgr_tree where emplid = '11528978'
;
commit
;
insert into ps_ry_lnmgr_tree
select
'11528978'
, 1
, 1
,
'<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">' ||
chr(10) ||
'<html xmlns="http://www.w3.org/1999/xhtml">' || chr(10) ||
'<head>' || chr(10) ||
'<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />' || chr(10) ||
'<title>Line Manager Connection</title>' || chr(10) ||
'<style type="text/css">' || chr(10) ||
'.box {' || chr(10) ||
'    text-align: center;' || chr(10) ||
'    vertical-align: middle;' || chr(10) ||
'    color: white;' || chr(10) ||
'    border: 1px solid #000000;' || chr(10) ||
'    padding: 1px 4px;' || chr(10) ||
'    background-color: red; ' || chr(10) ||
'    font-family: "Times New Roman", Times, serif;' || chr(10) ||

```

```

'    font-size: small;'                || chr(10) ||
'    vertical-align :middle;'          || chr(10) ||
'    width: 120px;'                    || chr(10) ||
'    height: 45px;'                    || chr(10) ||
'    font-weight:bold;'                || chr(10) ||
'    -ms-filter: "progid:DXImageTransform.Microsoft.Glow(color=#666666,strength=1)
progid:DXImageTransform.Microsoft.Shadow(color=#000000,direction=100,strength=4)";' || chr(10) ||
'    filter: progid:DXImageTransform.Microsoft.Glow(color=#cccccc,strength=1)
progid:DXImageTransform.Microsoft.Shadow(color=#000000,direction=150,strength=8);' || chr(10) ||
'    box-shadow: 3px 3px 3px #000;'    || chr(10) ||
'                                      || chr(10) ||
'                                      || chr(10) ||
'.blank_box'                          || chr(10) ||
'{'                                    || chr(10) ||
'    border-top-style:none;'            || chr(10) ||
'    border-top-color:white;'           || chr(10) ||
'    border-bottom-style:none;'         || chr(10) ||
'    border-bottom-color:white;'        || chr(10) ||
'    border-left-style:none;'           || chr(10) ||
'    border-left-color:white;'          || chr(10) ||
'    border-right-style:none;'          || chr(10) ||
'    border-right-color:white'          || chr(10) ||
'    width:160px'                       || chr(10) ||
'}'                                     || chr(10) ||
'.top_connector'                       || chr(10) ||
'{'                                     || chr(10) ||
'    border-top-style:none;'            || chr(10) ||
'    border-top-color:white;'           || chr(10) ||
'    border-bottom-style:none;'         || chr(10) ||
'    border-bottom-color:white;'        || chr(10) ||
'    border-left-style:none;'           || chr(10) ||

```

```

'      border-left-color:white;'      || chr(10) ||
'      border-right-style:solid;'      || chr(10) ||
'      border-right-width:thin;'      || chr(10) ||
'      border-right-color:black;'      || chr(10) ||
'      width:20px'                    || chr(10) ||
'}'                                    || chr(10) ||
'.bottom_connector'                   || chr(10) ||
{'                                     || chr(10) ||
'      border-top-style:none;'          || chr(10) ||
'      border-top-color:white;'         || chr(10) ||
'      border-bottom-style:solid;'      || chr(10) ||
'      border-bottom-width:thin;'      || chr(10) ||
'      border-bottom-color:black;'      || chr(10) ||
'      border-left-style:none;'         || chr(10) ||
'      border-left-color:white;'        || chr(10) ||
'      border-right-style:none;'        || chr(10) ||
'      border-right-color:white;'       || chr(10) ||
'      width:20px'                     || chr(10) ||
'}'                                    || chr(10) ||
'.top_bottom_connector'                || chr(10) ||
{'                                     || chr(10) ||
'      border-top-style:none;'          || chr(10) ||
'      border-top-color:white;'         || chr(10) ||
'      border-bottom-style:solid;'      || chr(10) ||
'      border-bottom-width:thin;'      || chr(10) ||
'      border-bottom-color:black;'      || chr(10) ||
'      border-left-style:none;'         || chr(10) ||
'      border-left-color:white;'        || chr(10) ||
'      border-right-style:solid;'       || chr(10) ||
'      border-right-width:thin;'        || chr(10) ||

```

```
'      border-right-color:black;' || chr(10) ||
'      width:20px'                || chr(10) ||
'}'                                || chr(10) ||
'.blank_cell'                      || chr(10) ||
'{ '                                || chr(10) ||
'      width:20px;'                || chr(10) ||
'}'                                || chr(10) ||
'.cell_separator'                  || chr(10) ||
'{ '                                || chr(10) ||
'      width:20px;'                || chr(10) ||
'}'                                || chr(10) ||
'.div-style{'                      || chr(10) ||
' visibility:hidden;'              || chr(10) ||
' background-color:lightblue;'     || chr(10) ||
' color:black;'                    || chr(10) ||
' font-weight:bold;'               || chr(10) ||
' border-bottom:thin;'             || chr(10) ||
' border-bottom-style:solid;'      || chr(10) ||
' border-bottom-color:black;'      || chr(10) ||
' border-top:thin;'                || chr(10) ||
' border-top-style:solid;'          || chr(10) ||
' border-top-color:black;'          || chr(10) ||
' border-left:thin;'               || chr(10) ||
' border-left-style:solid;'         || chr(10) ||
' border-left-color:black;'         || chr(10) ||
' border-top:thin;'                || chr(10) ||
' border-top-style:solid;'          || chr(10) ||
' border-top-color:black;'          || chr(10) ||
' border-right:thin;'              || chr(10) ||
' border-right-style:solid;'        || chr(10) ||
```

' border-right-color:black;'	chr(10)
' width:200px;'	chr(10)
' overflow:auto;'	chr(10)
'}'	chr(10)
'</style>'	

from dual

;

commit

;

set sqlterminator ']'

 -- 9.Generate HTML for Javascript functions

insert into ps_ry_lnmgr_tree

select /* create <table> tag */

'11528978'

, 1

, 2

, '<script language="javascript">

function Get_More(Elem_ID)

{

var obj = document.getElementById(Elem_ID);

var curleft = 0;

var curtop = 0;

var curheight = obj.offsetHeight/3;

if(obj.offsetParent)

while(1)

{

curleft += obj.offsetLeft;

curtop += obj.offsetTop + curheight;

```

        if(!obj.offsetParent)
            break;
        obj = obj.offsetParent;
    }
    else if(obj.x)
        curtop += obj.y + curheight
    if(obj.y)
        curtop += obj.y;
    document.getElementById(Elem_ID+"-more").style.visibility = "visible";
    document.getElementById(Elem_ID+"-more").style.position = "absolute";
    document.getElementById(Elem_ID+"-more").style.top = curtop + "px";
    document.getElementById(Elem_ID+"-more").style.left = curleft + "px";
}
function Get_Less(Elem_ID)
{
    var obj = document.getElementById(Elem_ID);
    document.getElementById(Elem_ID+"-more").style.visibility = "hidden";
}
function Get_GoogleMap(Elem_ID)
{
    var obj = document.getElementById(Elem_ID);
    pop_link = "https://maps.google.com/?daddr=" + obj.innerHTML.replace(/<BR>\\*/g, " ");
    var curleft = 0;
    var curtop = 0;
    if(obj.offsetParent)
        while(1)
        {
            curleft += obj.offsetLeft + 68;
            curtop += obj.offsetTop - 20;
            if(!obj.offsetParent)

```

```

        break;
        obj = obj.offsetParent;
    }
    else if(obj.x)
        curleft += obj.x + 68;
    if(obj.y)
        curtop += obj.y;
    document.getElementById("div_pop").innerHTML = "<table width='800' height='600'> <tr id='close_div'
    style='background:white;color:red;font-weight:bold' onclick=Hide_GoogleMap()>X - Click to close</tr><tr><iframe src='\" + pop_link + '\"
    border='1' width='100%' height='500'></iframe></tr></table>";
    document.getElementById("div_pop").style.visibility = "visible";
    document.getElementById("div_pop").style.position = "absolute";
    document.getElementById("div_pop").style.top = curtop + "px";
    document.getElementById("div_pop").style.left = curleft + "px";
}
function Hide_GoogleMap()
{
    document.getElementById("div_pop").innerHTML="";
    document.getElementById("div_pop").style.visibility = "hidden";
}
</script>
</head>
<body>' html
from dual
]
set sqlterminator ';'
commit
;
-----
-- 10 .Generate Title and User Instruction

```

```

-----
insert into ps_ry_lnmgr_tree
select /* Page Heading */
'11528978'
, 1
, 3
, '<p align="center">
<span style="font-size:x-large; font-family:"Juice ITC"; font-weight:bold; color:red">Line Manager Connection</span>
<br />
<span style="font-size:small;font-weight:bold">Click on box to show details</span>
<br />
<span style="font-size:small;font-weight:bold">Double-click on box to hide details </span>
</p>'
from dual
;
commit
;

```

```

-----
-- 11. Create <table>. Create 1st row using level 1 data (i.e. set width to max_cnt * 20 px)
-----

```

```

insert into ps_ry_lnmgr_tree
select
'11528978'
, 1
, 4
, '<table cellSpacing="0" cellPadding="0" align="center" width="" || (select max_cnt * 20 from ps_ry_tree_wrk a where a.mylevel = 1) || 'px">' ||
'<tr style="width:' || (select max_cnt * 20 from ps_ry_tree_wrk a where a.mylevel = 1) || 'px">' ||
'<td colspan="" || (select max_cnt from ps_ry_tree_wrk a where a.mylevel = 1) || ""></td></tr>'
from dual
;

```

```
commit
```

```
;
```

```
-----  
-- 12. Create Box for parent of parent, if applicable  
-----
```

```
insert into ps_ry_lnmgr_tree
```

```
select
```

```
'11528978'
```

```
, 20
```

```
, rownum
```

```
, a.html
```

```
from
```

```
(
```

```
select /* parent of parent box */
```

```
    case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr  
style="height:20px">' else " end ||
```

```
    /* boxes */
```

```
    case when a.parent_id = '00000000' then '<td class="blank_box" colspan="8"></td>'
```

```
    else
```

```
        case when a.max_cnt = 8 then
```

```
            '<td id="" || a.parent_id || "" colspan="6" class="box" onclick="Get_More(this.id)" ondblclick="Get_Less(this.id)">' ||
```

```
a.parent_name || '</td>' ||
```

```
            '<td class="cell_separator"></td><td class="cell_separator"></td>'
```

```
        else
```

```
            '<td colspan="" || (a.mid_point -3) || "" style="width:' || ((a.mid_point -3) *20) || 'px;"></td>' ||
```

```
            '<td id="" || a.parent_id || "" colspan="6" class="box" onclick="Get_More(this.id)" ondblclick="Get_Less(this.id)">' ||
```

```
a.parent_name || '</td>' ||
```

```
            '<td colspan="" || (a.mid_point -1) || "" style="width:' || ((a.mid_point -1) *20) || 'px;"></td>'
```

```
        end
```

```
    end ||
```

```

/* </tr> tag */
case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else "
end html
from ps_ry_tree_wrk a
where a.mylevel = 1
and exists (select 'x' from ps_employees b where b.emplid = a.parent_id)
) a
;
commit
;
-----
-- 13. Create connector for Level 1 to parent, if applicable
-----
insert into ps_ry_lnmgr_tree
select /* create <table> tag */
'11528978'
, 30
, rownum
, a.html
from
(
select
case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr
style="height:20px">' else " end ||
case when a.id = '00000000' then '<td class="blank_box" colspan="8"></td>'
else
case when a.max_cnt = 8 then
'<td colspan="2" style="width:40px;"></td><td class="top_connector"></td><td colspan="3" style="width:60px;"></td>' ||
'<td class="blank_cell"></td><td class="blank_cell"></td>'
else

```

```

        '<td colspan="' || (a.mid_point - 3) || '" style=width:"' || ((a.mid_point - 3) * 20) || 'px;"></td>' ||
        '<td colspan="2" style="width:40px;"></td><td class="top_connector"></td><td colspan="3" style="width:60px;"></td>' ||
        '<td colspan="' || (a.mid_point - 1) || '" style=width:"' || ((a.mid_point - 1) * 20) || 'px;"></td>'
    end
end ||
case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else " end
html
from ps_ry_tree_wrk a
where a.mylevel = 1 and a.parent_id <> '' and a.parent_name <> ''
) a
;
commit
;
-----
-- 14. Create Level 1 box
-----
insert into ps_ry_lnmgr_tree
select /* create <table> tag */
'11528978'
, 32
, rownum
, a.html
from
(
select
/* generate <tr> tag */
case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr>' else " end ||
/* boxes */
case when a.id = '00000000' then '<td class="blank_box" colspan="8"></td>'
else

```

```

        case when a.max_cnt = 8 then
            '<td id="' || a.id || '" colspan="6" class="box" onclick="Get_More(this.id)" ondblclick="Get_Less(this.id)">' || a.id_name || '</td>'
        ||
            '<td class="cell_separator"></td><td class="cell_separator"></td>'
        else
            '<td colspan="' || (a.mid_point - 3) || '" style="width:' || ((a.mid_point - 3) * 20) || 'px;"></td>' ||
            '<td id="' || a.id || '" colspan="6" class="box" onclick="Get_More(this.id)" ondblclick="Get_Less(this.id)">' || a.id_name || '</td>' ||
            '<td colspan="' || (a.mid_point - 1) || '" style="width:' || ((a.mid_point - 1) * 20) || 'px;"></td>'
        end
    end ||
    /* generate </tr> tag */
    case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else " end
    html
    from ps_ry_tree_wrk a
    where a.mylevel = 1
    ) a
    ;
    commit
    ;
    -----
    -- 15. Create Level 1 box bottom horizontal connector
    -----
    insert into ps_ry_lnmgr_tree
    select /* create <table> tag */
    '11528978'
    , 40
    , rownum
    , a.html
    from
    (

```

```

select
/* generate <tr> tag */
case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr
style="height:45px">' else " end ||
case when a.id = '00000000' then '<td class="blank_box" colspan="" || a.max_cnt || ""></td>'
else
case when a.child_cnt = 1 then
case when (select b.id from ps_ry_tree_wrk b
where b.mylevel = a.mylevel + 1
and b.parent_id = a.id) = '00000000' then '<td class="blank_box" colspan="8"></td>'
else '<td class="blank_cell"></td><td class="blank_cell">' ||
'<td class="top_connector"></td>' ||
'<td class="blank_cell"></td><td class="blank_cell"></td><td class="blank_cell"></td><td class="blank_cell"></td><td
class="blank_cell"></td>'
end
else
-- 1st child left offset from mid point
'<td class="blank_cell" colspan="" ||
(select (b.mid_point) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.first_child = 'Y')
|| ""></td>' ||
-- 1st child right offset
'<td class="bottom_connector" colspan="" || ((a.mid_point -2) -
(select (b.mid_point -1) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.first_child =
'Y')) || ""></td>' ||
-- mid point of previous Level
'<td class="top_bottom_connector"></td>' ||
-- right offset from midpoint
'<td class="bottom_connector" colspan="" || (a.max_cnt -
(select (b.mid_point) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.first_child = 'Y') -

```

```

        (select (((b.max_cnt)/2)+1)   from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.last_child =
'Y') -
        ((a.mid_point -1) -
        (select (b.mid_point) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.first_child =
'Y')) - 1) || "'></td>' ||
-- last child right
        '<td class="blank_cell" colspan="' ||
        (select (b.mid_point +2) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.last_child =
'Y') || "'></td>'
        end
end ||
/* generate </tr> tag */
case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else " end
html
from ps_ry_tree_wrk a
where a.mylevel = 1
) a
;
commit
;
-----
-- 16. create vertical connector from Level 2 to Level 1
-----
insert into ps_ry_lnmgr_tree
select /* create <table> tag */
'11528978'
, 41
, rownum
, a.html
from

```

```

(
select
case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr
style="height:20px">' else " end ||
case when a.id = '00000000' then '<td class="blank_box" colspan="8"></td>'
else
case when a.max_cnt = 8 then
'<td colspan="2" style="width:40px;"></td><td class="top_connector"></td><td colspan="3" style="width:60px;"></td>' ||
'<td class="blank_cell"></td><td class="blank_cell"></td>'
else
'<td colspan="' || (a.mid_point -3) || '" style="width:" || ((a.mid_point -3) *20) || 'px"></td>' ||
'<td colspan="2" style="width:40px;"></td><td class="top_connector"></td><td colspan="3" style="width:60px;"></td>' ||
'<td colspan="' || (a.mid_point -1) || '" style="width:" || ((a.mid_point -1) *20) || 'px"></td>'
end
end ||
case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else " end
html
from ps_ry_tree_wrk a
where a.mylevel = 2
)
a
;
commit
;
-----
-- 17. create Level 2 boxes
-----
insert into ps_ry_lnmgr_tree
select /* create <table> tag */
'11528978'

```

```

, 42
, rownum
, a.html
from
(
select
/* generate <tr> tag */
case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr>' else " end ||
/* boxes */
case when a.id = '00000000' then '<td class="blank_box" colspan="8"></td>'
else
case when a.max_cnt = 8 then
'<td id="" || a.id || "" colspan="6" class="box" onclick="Get_More(this.id)" ondblclick="Get_Less(this.id)">' || a.id_name || '</td>'
||
'<td class="cell_separator"></td><td class="cell_separator"></td>'
else
'<td colspan="" || (a.mid_point - 3) || "" style="width:' || ((a.mid_point - 3) * 20) || 'px;"></td>' ||
'<td id="" || a.id || "" colspan="6" class="box" onclick="Get_More(this.id)" ondblclick="Get_Less(this.id)">' || a.id_name || '</td>' ||
'<td colspan="" || (a.mid_point - 1) || "" style="width:' || ((a.mid_point - 1) * 20) || 'px;"></td>'
end
end ||
/* generate </tr> tag */
case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else " end
html
from ps_ry_tree_wrk a
where a.mylevel = 2
) a
;
commit
;

```

```
-----  
-- 18. create Level 2 boxes bottom horizontal connector  
-----
```

```
insert into ps_ry_lnmgr_tree  
select /* create <table> tag */  
'11528978'  
, 50  
, rownum  
, a.html  
from  
(  
select  
/* generate <tr> tag */  
case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr  
style="height:45px">' else " end ||  
case when a.id = '00000000' then '<td class="blank_box" colspan="" || a.max_cnt || ""></td>'  
  else  
    case when a.child_cnt = 1 then  
      case when (select b.id from ps_ry_tree_wrk b  
        where b.mylevel = a.mylevel + 1  
        and b.parent_id = a.id) = '00000000' then '<td class="blank_box" colspan="8"></td>'  
      else '<td class="blank_cell"></td><td class="blank_cell">' ||  
        '<td class="top_connector"></td>' ||  
        '<td class="blank_cell"></td><td class="blank_cell"></td><td class="blank_cell"></td><td  
class="blank_cell"></td>'  
      end  
    else  
-- 1st child left offset from mid point  
      '<td class="blank_cell" colspan="" ||
```

```

                (select (b.mid_point) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.first_child = 'Y')
|| ""></td>' ||
-- 1st child right offset
        '<td class="bottom_connector" colspan="" || ((a.mid_point -2) -
                (select (b.mid_point -1) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.first_child =
'Y')) || ""></td>' ||
-- mid point of previous Level
        '<td class="top_bottom_connector"></td>' ||
-- right offset from midpoint
        '<td class="bottom_connector" colspan="" || (a.max_cnt -
                (select (b.mid_point) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.first_child = 'Y') -
                (select (((b.max_cnt)/2)+1)    from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.last_child =
'Y') -
                ((a.mid_point -1) -
                (select (b.mid_point) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.first_child =
'Y')) - 1) || ""></td>' ||
-- last child right
        '<td class="blank_cell" colspan="" ||
                (select (b.mid_point +2) from ps_ry_tree_wrk b where b.mylevel = a.mylevel+1 and b.parent_id = a.id and b.last_child =
'Y') || ""></td>'
        end
end ||
/* generate </tr> tag */
case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else " end
html
from ps_ry_tree_wrk a
where a.mylevel = 2
) a
;
commit

```

```

;
-----
-- 19. create vertical connector from Level 3 to Level 2
-----

insert into ps_ry_lnmgr_tree
select /* create <table> tag */
'11528978'
, 51
, rownum
, a.html
from
(
select
case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr
style="height:20px">' else " end ||
case when a.id = '00000000' then '<td class="blank_box" colspan="8"></td>'
else
case when a.max_cnt = 8 then
'<td colspan="2" style="width:40px;"></td><td class="top_connector"></td><td colspan="3" style="width:60px;"></td>' ||
'<td class="blank_cell"></td><td class="blank_cell"></td>'
else
'<td colspan="3" || (a.mid_point -3) || "' style=width:" || ((a.mid_point -3) *20) || 'px"></td>' ||
'<td colspan="2" style="width:40px;"></td><td class="top_connector"></td><td colspan="3" style="width:60px;"></td>' ||
'<td colspan="3" || (a.mid_point -1) || "' style=width:" || ((a.mid_point -1) *20) || 'px"></td>'
end
end ||
case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else " end
html
from ps_ry_tree_wrk a
where a.mylevel = 3

```

```

) a
;
commit
;
-----
-- 20. create Level 3 boxes
-----

insert into ps_ry_lnmgr_tree
select /* create <table> tag */
'11528978'
, 52
, rownum
, a.html
from
(
select
/* generate <tr> tag */
case when lag(a.mylevel) over (order by a.mylevel) is null or lag(a.mylevel) over (order by a.mylevel) <> a.mylevel then '<tr>' else " end ||
/* boxes */
case when a.id = '00000000' then '<td class="blank_box" colspan="8"></td>'
else
case when a.max_cnt = 8 then
'<td id="' || a.id || '" colspan="6" class="box" onclick="Get_More(this.id)" ondblclick="Get_Less(this.id)">' || a.id_name || '</td>'
||
'<td class="cell_separator"></td><td class="cell_separator"></td>'
else
'<td colspan="' || (a.mid_point - 3) || '" style="width:' || ((a.mid_point - 3) * 20) || 'px;"></td>' ||
'<td id="' || a.id || '" colspan="6" class="box" onclick="Get_More(this.id)" ondblclick="Get_Less(this.id)">' || a.id_name || '</td>' ||
'<td colspan="' || (a.mid_point - 1) || '" style="width:' || ((a.mid_point - 1) * 20) || 'px;"></td>'
end
end

```

```

end ||
/* generate </tr> tag */
case when lead(a.mylevel) over (order by a.mylevel) is null or lead(a.mylevel) over (order by a.mylevel) <> a.mylevel then '</tr>' else " end
html
from ps_ry_tree_wrk a
where a.mylevel = 3
) a
;
commit
;
-----
-- 21. Generate close table </table> tag
-----
insert into ps_ry_lnmgr_tree
select /* create <table> tag */
'11528978'
, 90
, rownum
, '</table>'
from dual
;
commit
;
-----
-- 22. Generate employee photo, contact details for the onclick function
-----
insert into ps_ry_lnmgr_tree
select /* create <table> tag */
'11528978'
, 91

```

```

, rownum
, a.html
from
(
select /*+ index_rs_asc(a) */ /*+ leading(a) */ '<div id="" || a.id || '-more" class="div-style">' ||
(select case when get_ps_empl_photo(b.emplid) is null then (select GET_PS_EMPL_PHOTO2('SANTANDER_RED_FLAME_LOGO')
from dual)
else get_ps_empl_photo(b.emplid)
end
from dual) || chr(10) ||
'<br />' || a.id_name || '<br />' || b.jobtitle || '<br />Department : ' || b.deptname || chr(10) ||
coalesce((select '<br /><span id="G_ID_' || a.id || '" onclick="Get_GoogleMap(this.id)">' ||
l.address1 || '<br />' || l.address2 || '<br />' || l.city || '<br />' || l.state || '<br />' || l.postal || '</span>' from ps_location_tbl l
where l.setid = 'COMMN'
and l.location = b.location
and l.effdt = (select max(l2.effdt) from ps_location_tbl l2
where l2.setid = l.setid
and l2.location = l.location
and l2.effdt <= b.effdt)
),') ||
coalesce((select '<br />Contact : ' || p.phone from ps_personal_phone p where p.emplid = b.emplid and p.phone_type = 'BUSN'),') || '</div>'
html
from ps_ry_tree_wrk a, ps_employees b
where a.id = b.emplid
and a.id <> '00000000'
union all
select /*+ leading(a) index_rs_asc(a) */ '<div id="" || a.parent_id || '-more" class="div-style">' ||
(select case when get_ps_empl_photo(a.parent_id) is null then (select GET_PS_EMPL_PHOTO2('SANTANDER_RED_FLAME_LOGO')
from dual)
else get_ps_empl_photo(a.parent_id)

```

```

        end
    from dual) ||
'<br />' || a.parent_name || '<br />' || b.jobtitle || '<br />' || b.deptname ||
    coalesce((select '<br /><span id="G_ID_' || a.parent_id || '" onclick="Get_GoogleMap(this.id)">*' ||
l.address1 || '<br />' || l.address2 || '<br />' || l.city || '<br />' || l.state || '<br />' || l.postal || '</span>' from ps_location_tbl l
    where l.setid = 'COMMN'
    and l.location = b.location
    and l.effdt = (select max(l2.effdt) from ps_location_tbl l2
        where l2.setid = l.setid
        and l2.location = l.location
        and l2.effdt <= b.effdt)
    ),'') ||
    coalesce((select '<br />Contact : ' || p.phone from ps_personal_phone p where p.emplid = b.emplid and p.phone_type = 'BUSN'),'') || '</div>'
html
from ps_ry_tree_wrk a, ps_employees b
where b.emplid = a.parent_id
    and a.mylevel = 1
) a
;
commit
;
-----
-- 23. This generates a generic container for holding the employee details.
--    Element Id is div_pop and defaults to invisible until the chart box is clicked
-----
insert into ps_ry_lnmgr_tree
select
'11528978'
, 99
, rownum

```

```
, '<div id="div_pop" style="visibility:hidden">
</div>
</body>
</html>'
from dual
;
commit
;
set pagesize 0 termout off
spool c:\connection_tree.htm
select tree_html from ps_ry_lnmgr_tree where emplid = '11528978' order by emplid, seq, seq2
;
spool off
;
set feedback on echo on define & sqlterminator ';' wrap on linesize 999 pagesize 999 trimspool on LONGCHUNKSIZE 1000000 long 1000000
termout on
```