SQL Query Examples

Andrea Brice

Tuesday, May 19, 2015

Purpose

The purpose of this document is to demonstrate some of the code I created for reporting off of various databases. I keep sample / snippets of code when I learn something new so I can reuse the syntax in the future. The schema, some of the logic, the names, and other pieces of information have been changed so this work is non-functional, for example only.

I had more flexibility in code as we went through the Oracle upgrades to version 12. Some of the statistical analysis statements were only available to me after the most recent upgrade on one of the databases.

Also, I come to this work from an engineering perspective. I keep track of my syntax by keeping copies of useful code and pasting that into work as I need. One of my most used statements, the rank statement, has a structure I continue to forget to this day because I simply "copy" / "paste" it into code so often I never remembered.

Another thing you'll find is much of my parsing is not done with REGEXP because I did not have access to that group of commands until recently, and even then, not across all databases. So much of my older code used INSTR, SUBSTR, etc. I have, however, included some code with examples of using REGEXP syntax.

Code Example 1 - nested queries, ranking, creating key fields / unique identifiers, joining, dblink

This bit of code creates a view based off dblinked connection to another database for configuration information. The challenges in joining the two tables is that the data stream is often interrupted, so I have to query across the report period range and pull only the latest configuration data. I use row_number for the rank and partition by a variety of attributes. In this case, the order is only by one attribute, but I do have queries where there are more than one attribute in both the partition and the order by statements of row_number. Besides struggling with inconsistent reporting and a fixed date range to report by, I'm having to deal with inconsistent data types, character strings, and creating unique identifiers for key field creation.

---

create or replace view MSC_SUBNET_CONNECTIONS as

WITH

DATES AS (SELECT DATE1 - 7 as date1 FROM CIQ_DATES – SELECT trunc(sysdate-7) as date1 from dual)

,SUBNET_BASE AS (select * from (select

```
upper(SWITCH) as msc,
row_number () over (partition by switch, subnet_base, subnet_number order by
```

start_date desc) as row_number, upper(SWITCH)||'-'||SUBNET_NUMBER as msc_subnet_num, upper(SWITCH)||'-'||SUBNET_BASE as msc_subnet_base, SUBSTR(SUBNET_BASE,1,INSTR(SUBNET_BASE, '.',1,3)) AS SUBNET_STRING, SUBSTR(SUBNET_BASE, INSTR(SUBNET_BASE, '.',1,3)+1,LENGTH(SUBNET_BASE)-INSTR (SUBNET_BASE, '.',1,3))*1 AS STR_LASTNUM, SUBNET_NUMBER, SUBNET_BASE, SUBNET_BASE_NUM, SUBNET_MASK, SUBNET_MASK_NUM, SUBNET_NAME

```
        from  XXX.ADM_EXT_SUBNET@otherdb, dates
        where trunc(start_date) > = date1
```

) where row_number = 1)

,OFFSET AS (select * from (select row_number () over (partition by switch, subnet_number,EXT_OFFSET_NUMBER order by start_date desc) as row_number, upper(SWITCH)||'-'||SUBNET_NUMBER as msc_subnet_num, upper(SWITCH)||'-'||EXT_OFFSET_NUMBER AS MSC_OFFSET, upper(SWITCH) as msc, start_date, SUBNET_NUMBER, EXT_OFFSET_NUMBER*1 AS EXT_OFFSET_NUMBER, EXT_OFFSET_NAME

```
    from XXX.ADM_EXT_OFFSET@otherdb, dates
        where trunc(start_date) > = date1

     )     where row_number = 1)

    SELECT
        A.msc_subnet_num,
        A.MSC_OFFSET,
        B.msc_subnet_base,
        A.MSC,
```

– B.SUBNET_BASE, B.SUBNET_NAME, A.EXT_OFFSET_NAME, B.SUBNET_STRING||(B.STR_LASTNUM+A.EXT_ AS NEWSTR, B.SUBNET_BASE_NUM, B.SUBNET_MASK, B.SUBNET_MASK_NUM, – B.SUBNET_NUMBER,
– B.SUBNET_STRING, – B.STR_LASTNUM, A.EXT_OFFSET_NUMBER

```
    FROM OFFSET A LEFT JOIN SUBNET_BASE B
    ON A.msc_subnet_num = B.msc_subnet_num
```

Code Example 2 - use of REGEXP and converting between decimal, hex, and binary, crossing datamarts

This snippet of code demonstrates the use of REGEXP_SUBSTR to parse out portions of information from within a single field.

CREATE OR replace FORCE VIEW MGW_LNG_MAP AS SELECT a.MSC_LNG AS SWITCH_LNG, MSC_MGWOBJ AS SWITCH_MGW_OBJ, REGEXP_SUBSTR(a.MSC_LNG, '[[:alnum:]]+', 1, 1,'i') as switch, NMSC AS MSC, REGEXP_SUBSTR (a.MSC_LNG, '[0-9]+', 1, 2)AS LNG, REGEXP_SUBSTR (OMSC_MGWOBJ,'[0-9]+', 1,2)AS MGW_OBJ, NMSC|| '-'||REGEXP_SUBSTR (a.MSC_LNG, '[0-9]+', 1, 2) AS MSC_LNG, NMSC|| '-'|| REGEXP_SUBSTR (MSC_MGWOBJ,'[0-9]+', 1,2) aS MSC_MGW_OBJ

```
 FROM TEST.MGW_LNG_MAPPING a LEFT JOIN TEST.MSC_NAMES_2SW
        ON REGEXP_SUBSTR(a.MSC_LNG, '[[:alnum:]]+', 1, 1,'i') = OMSC;
```

Parsing both numeric and char data. Also joining data across multiple datamarts, multiple vendors, converting across multiple datatypes - not just char to numeric, but dec to hex - concatenating data to create multiple versions for this view for future joining, ranking, and joining all this data into a new standardized, single view.

```
create or replace view test.active_ID_INFO_base AS
with dates as (select
    * from PROD_SCHEMA.NU5DATES
)

,NO_base as (
  select
    row_number () over (partition by ELEMENT_NAME, SP_CODE_HD, pulldate order by pulltime desc) as r
    to_char(pulldate, 'DD-MON-YYYY') as dow,
    pulltime,
    ELEMENT_NAME as GS,
    ELEMENT_TYPE,
    SP_CODE_HD as original_data,
    substr(SP_CODE_HD,1,instr(SP_CODE_HD, '/',1,1)-1) as hex_pc,
    regexp_substr(SP_CODE_HD,'[^/]+',1,2) as  orig_pc
  from NO.ZNRI_GS_NET@otherdb, dates
  where pulldate between date1 and date3
  --and ELEMENT_NAME like 'S%'
       )
,NO_GS as (
      select
        row_number,
        dow,
        GS,
        'NO' as vendor,
        '' as e164,
        original_data,
        hex_pc,
        to_number(hex_pc, 'XXXXXXXXXX') as decver_pc,
        orig_pc,
        replace(TO_CHAR(to_number(substr(hex_pc,1,2), 'XXX'),'099')||'-'||TO_CHAR(to_number(substr(h
dash_pc_long,
        replace(TO_CHAR(to_number(substr(hex_pc,1,2), 'XXX'),'099')||'.'||TO_CHAR(to_number(substr(h
        replace(TO_CHAR(to_number(substr(hex_pc,1,2), 'XXX'),'999')||'-'||TO_CHAR(to_number(substr(h
        replace(TO_CHAR(to_number(substr(hex_pc,1,2), 'XXX'),'999')||'.'||TO_CHAR(to_number(substr(h
-- replace(dash_pc_long, '-','.') as dot_pc_long,
--   REGEXP_SUBSTR(dash_pc_long, '[0-9]+', 1, 1)*1||'.'||REGEXP_SUBSTR(dash_pc_long, '[0-9]+', 1, 2)*1
 --  REGEXP_SUBSTR(dash_pc_long, '[0-9]+', 1, 1)*1||'-'||REGEXP_SUBSTR(dash_pc_long, '[0-9]+', 1, 2)*1
from NO_base
where row_number = 1

  )

,E_pc as (
        select * from (
          select
                row_number () over (partition by managedelement_id, start_date order by start_time d
```

```
                to_char(start_date, 'DD-MON-YYYY') as dow,
            managedelement_id as GS,
            'E' as vendor,
            SPC8BITFORMAT as dash_pc,

            REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 1)*1||'.'||REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+',

            replace(
            to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 1), '099')||'.'||
            to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 1), '099')||'.'||
            to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 3), '099'),' ','') as dot_pc_long,
            replace(
            to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 1), '099')||'-'||
            to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 1), '099')||'-'||
            to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 3), '099'),' ','')  as dash_pc_long,

            --opc as decver_pc,
            replace(to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 1)*1,'XX'),' ','')||
            replace(to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 2)*1,'XX'),' ','')||
            replace(to_char(REGEXP_SUBSTR(SPC8BITFORMAT, '[0-9]+', 1, 3)*1,'XX'),' ','') as hex_pc

    from E_UMTS.UN_OCAL_SIGNAL_PT@otherdb, dates
    where start_date between date1 and date3
) where row_number = 1
 )
 ,E_e164 as (
select * from (
 select
     row_number () over (partition by managedelement_id, start_date order by start_time desc) as row
     to_char(start_date, 'DD-MON-YYYY') as dow,
     managedelement_id as GS,
     REGEXP_SUBSTR(addr, '[0-9]+', 1, 1) as e164


 from E_UMTS.GS_CONFIG_SCCP_GTRULE@otherdb, dates
 where start_date between date1 and date3
     and SCCP_GTTRANS_ID = 'GtTranslator_E164' and addr <> '*'
 ) where row_number = 1

 )

 ,ERIC AS (
    select
      a.dow,
         a.GS,
         vendor,
         e164,
         dash_pc,
         dot_pc,
         dot_pc_long,
         dash_pc_long,
         to_number(hex_pc, 'XXXXXXXXXX') as decver_pc,
         hex_pc
     from E_pc a left join E_e164 b on a.GS= b.GS and a.dow = b.dow
```

```
    )

SELECT dow,GS,vendor,e164,dash_pc,dot_pc,dot_pc_long,dash_pc_long,decver_pc,hex_pc FROM NO_GS
UNION ALL
SELECT dow,GS,vendor,e164,dash_pc,dot_pc,dot_pc_long,dash_pc_long,decver_pc,hex_pc FROM ERIC

ORDER BY DOW, GS
```

Code Example #3 - Use of PIVOT to count records reporting to list of tables.

Count the number of records, and / or nodes reporting over a date range using the PIVOT function

---

```
with
      datum as (
                  SELECT DISTINCT mgw, datetime, to_char(datetime, 'DD-MON-YYYY')  as dow,
      'ETHERNETSWITCHMOD' as table_name FROM EMMG.ETHERNETSWITCHMOD where TRUNC
      (DATETIME) > = TRUNC(SYSDATE) UNION ALL
                  SELECT DISTINCT mgw, datetime, to_char(datetime, 'DD-MON-YYYY')  as dow,
      'ETRESOURCE' as table_name FROM EMMG.ETRESOURCE where TRUNC(DATETIME) > =
      TRUNC(SYSDATE) UNION ALL
                  SELECT DISTINCT mgw, datetime, to_char(datetime, 'DD-MON-YYYY')  as dow,
      'GIGABITETHERNET' as table_name FROM EMMG.GIGABITETHERNET where TRUNC(DATETIME)
      > = TRUNC(SYSDATE) UNION ALL
                  SELECT DISTINCT mgw, datetime, to_char(datetime, 'DD-MON-YYYY')  as dow,
      'VMGW' as table_name FROM EMMG.VMGW where TRUNC(DATETIME) > = TRUNC(SYSDATE)

          )
          --where dow like '08-APR-2015'
       ,dataall as (
          select
          dow,
          datetime,
          mgw,
          table_name
          from datum)



          select *  from dataall
              pivot(
                  count(datetime)
              for table_name in (

                      '
                      'IMEXTANNOUNCEMENTS',
                      'PLUGINUNIT_SPLIT',
                      'REMOTESITE',
                      'SCTP',
```

```
                        'SCTPASSOCIATION'
                        '

                    )
                )
        order by mgw, dow
```

******

Example #4 Case statements

Use of packages as well as case statements

–drop view no_msc_capacity /*

create or replace view no_msc as with all_rec as (select distinct a.mgw_lo, a.switch,b.SWITCH_NAME as clli, b.SWITCH_VER, b.SW_TYPE
from (select distinct mgw_lo, switch from ab.no_MGW_VSP_CAPACITY union all select distinct mgw_lo, switch from ab.no_EQPCOUNTS)a left join ab.no_MSC_GENERAL_INFO b on a.switch = b.switch), level_1 as (select

```
        d.DOW,
        c.clli,
        c.switch,
        c.switch_ver,
        c.sw_type,
        case when d.mgw_lo is null then c.switch||'-LOCAL' else d.mgw_lo end as MGW_lo,
        case when d.mgw_loc is null then 'LOCAL' else d.mgw_loc end as mgw_loc,
        nvl(d.GWINV_VSP_CARD_CT,0) as GWINV_VSP_CARD_CT,
        nvl(d.IU_IP_CONTEXT,0) as IU_IP_CONTEXT,
        nvl(i.NONA_CHANNELS,0) as NONA_CHANNELS,
        nvl(i.NONA_ERLANGS, 0) as NONA_ERLANGS,
        nvl(i.A_CHANNELS, 0) as A_CHANNELS,
        nvl(i.A_ERLANGS,0) as A_ERLANGS

 from all_rec c
    left join ab.no_VSP_SUMMED_CAPACITY d on  c.mgw_lo = d.mgw_lo
    left join ab.no_ANONA i on c.mgw_lo = i.MGW_lo
```

), LEVEL2 AS (

```
select
        case when aa.DOW is null then f.dow else aa.dow end as dow,
        o.region,
        o.market,
        aa.CLLI,
        aa.SWITCH,
        aa.SWITCH_VER,
```

```
aa.SW_TYPE,
 case when aa.GWINV_VSP_CARD_CT =0 or aa.switch = 'BXX' then 'TXX' else
        case when aa.MGW_LOC like 'REMOTE%' then 'RMGW' else 'CS/MGW' end end as NODE_TYPE,
case when sw_type = 'TYPEC' then 'OTHER' ELSE
    case when a_channels =0 and (ALLOC_AT_Contexts >0 or IP_CONTEXT > 0) then 'bb' else
        case when a_channels >0  and (ALLOC_AT_Contexts >0 or IIP_CONTEXT >0) then 'bc' else
            case when sw_type = 'CA' then 'gG' else 'cG' END END end end as FUNCTION,

o.MSC_FRIENDLY_NAME,
o.NETOPTS_CLLI,
o.POINT_CODE,
o.VENDOR,
o.PROCESSOR,

aa.MGW_lo,
aa.MGW_LOC,

aa.GWINSP_CARD_CT,
nvl(n.SIPI_P_COUNT,0) as SIPI_P_COUNT,

aa.MGW_CONTEXT_CAP,
trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,aa.MGW_CONTEXT_CAP)) AS MGW_CTXT_TRAFF_CAP,
aa.MGW_PORT_EQUIVALENT,

nvl(f.spm_dtc_eqpct_t1,0) as spm_dtc_t1,
nvl(f.spm_dtc_eqpct_chnl,0)  as dtc_spm_chnl,
trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,nvl(f.spm_dtc_eqpct_chnl,0))) AS SPM_DTC_EQM_TRA

aa.ATM_PVC_CT,

aa.ALLOC_ATM_Contexts,
aa.IU_IP_CONTEXT,
nvl(n.SIPI_CTX_CAP,0) as sipi_context_cap,
aa.ALLOC_TXX_MGW_CONTEXTS,


nvl(h.SPM_DTC_ALLOC_CHNL,0) as  SPM_DTC_ALLOC_CHNL,
aa.NONA_CHANNELS,
aa.A_CHANNELS,
aa.NONA_CHANNELS+aa.A_CHANNELS as alloc_TXX_chnl,


CASE WHEN aa.ALLOC_AT_Contexts= 0 THEN 0
    ELSE trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,aa.ALLOC_AT_Contexts)) END AS AT_TRAFF_

CASE WHEN aa.NONA_CHANNELS= 0 THEN 0
    ELSE trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,aa.NONA_CHANNELS)) END as nona_traff_ca
CASE WHEN aa.A_CHANNELS = 0 THEN 0
    ELSE trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,aa.A_ELS)) END as afc_traff_cap,
CASE WHEN aa.NONNELS+aa.A_CNELS = 0 THEN 0
    ELSE trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,aa.NONA_CHAS+aa.ANNELS)) END as tot_TXX

aa.A_RLAGS,
aa.NONA_RLNGS,
```

```
nvl(j.AT_ERL, 0) as aterl,
nvl(j.IP_ERL,0) as Ip_erl,
nvl(n.SIPRL,0) as sii_erl,
 aa.ALANGS+aa.NONA_ERLANGS AS TXX_ERL,
nvl(n.SIERL,0)+ nvl(j.IU_ERL,0)+nvl(j.A_ERL, 0)+A_ERL+ NONNGS as tot_erl_traff,


trunc(ac.erlangb_package.ERLANGB_capacity(0.xxx,nvl(n.SIERL,0)+ nvl(j.IUERL,0)+nvl(j.ATMERL, 0)+

ROUND(ALLOC_TEXTS/24,0) AS IUPort_alloc,
      ROUND(nvl(n.SICTX_CAP,0)/24,0) AS sipirt_alloc,
round(nvl(h.SPM_LLOC_CHNL,0)/24,0) as spm_d_alloc_t1
```

from level_1 aa left join ab.no_MSC_NAME o on aa.CLLI||'-'||aa.MGW_LOC = o.msc_mgw_lo
left join ab.no_RFF_SUM j on aa.mgw_lo = j.MGW_lo left join ab.no_SFF n on aa.mgw_lo = n.MGW_lo
left join ab.no_TXNNEL_CT h on aa.mgw_lo = h.MGW_lo left join ab.no_EQPCOUNTS f on aa.mgw_lo
= f.mgw_lo) SELECT DOW, region, market, CLLI, SWITCH, SWITCH_VER, SW_TYPE, NODE_TYPE,
FUNCTION, MSC_FRIENDLY_NAME, NETOPTS_CLLI, POINT_CODE, VENDOR, PROCESSOR,
MGW_lo,

```
    GWINV_VSP_CARD_CT,


    MGW_PORT_EQUIVALENT,
    SPC_T1,

    MG
    spm_dtc_alloc_t1,

   case when node_type = 'TXX' then MGW_PORT_EQUIVALENT+SPM_DTC_T1
    else MGW_PORT_EQUIVALENT+spm_dtc_alloc_t1 end as config_port_cap,
      QD_AVG_CONTEXTS,

    case when node_type = 'TXX' then MGW_CONTEXT_CAP+DTC_SPM_CHNL else
        MGW_CONTEXT_CAP+SPM_DTC_ALLOC_CHNL end as total_ctx_chnl_capacity,

    case when node_type = 'TXX'
            then trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,MGW_CONTEXT_CAP+DTC_SPM_CHNL))
            else trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,MGW_CONTEXT_CAP+SPM_DTC_ALLOC_CH

    round(TOT_ERL_TRAFF/case when node_type = 'TXX'
            then trunc(ac.erlangb_package.ERLANGB_OFFERED(0.xxx,MGW_CONTEXT_CAP+DTC_SPM_CHNL)) *
ROUND (A_ERLANGS / A_INTFC_TRAFF_CAP,3)*100 end AS a_intfc_resource_util,
```

– CASE WHEN NONA_TRAFF_CAP= 0 THEN 0 ELSE ROUND (NONA_

```
    FROM LEVEL2
    order by MGW_lo
```

SUM CASE statement

---

select a.mgw as mgw, a.mgw_lo as mgw_lo, sum(case when rncinf.signal_type = 'ATM' then 1 else 0 end) as ATM_RNC_count, sum(case when rncinf.signal_type = 'IUCS_IP'then 1 else 0 end) as IP_RNC_count, sum(case when rncinf.signal_type = 'ATM' then nvl(rncinf.PVC_PER_MGW,0) else 0 end) as atm_pvc_ct, sum(case when rncinf.signal_type = 'ATM' then nvl(rncinf.ATM_CHNL_PER_MGW,0) else 0 end) as atm_channels

```
from AB.MGWPOOL_INFO a left join AB.IVE_RNCS rncinf
on a.SW_NAME_ROUTESET = rncinf.MSC_ROUTESET_NAME
group by a.mgw_lo, a.MGW
order by mgw
```

Example #5 LAG and LEAD

Includes crossing multiple data marts, database versions, vendors, data types. Joins the normalized data into a single table so that all vendors' nodes (i.e., the whole network) are represented. The use of LAG and LEAD was applied for the vendor where there were individual rows of information and the range of one of the objects had to be aggregated so that the max and min number within the range were represented on a single line.

---

create or replace view MSC_DAC_CAC_RANGES AS

with dates as (select * from PROD.CF_DATES),

E as (select distinct DAC, switch, 'E' as vendor, min(cac) over (partition by switch, DAC, maxdiffcac) startrange, max(cac) over (partition by switch, DAC, maxdiffcac) endrange

from (select switch, DAC, cac, e, diffcac, max(diffcac) over (partition by switch, DAC order by cac RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) maxdiffcac

```
from (

         select switch, DAC, cac, e,
            sum(decode (diffcac, 1, 0, diffcac)) over (partition by switch, DAC order by cac RANGE
         from (

                select/*+DRIVING_SITE(a)*/  clli as switch, DAC, cac, LAG(cac, 1, 0) OVER (partitio
                        (cac - LAG(cac, 1, 0) OVER (partition by clli, DAC ORDER BY cac)) diffcac
                FROM n.DACcac@OTHERDB a, man.switches@OTHERDB b, dates
                where a.switch = b.switch
                and DAC >= 64000
                and pulldate >= date1 and pulldate < date3
         )
)
```

)), A AS (select distinct switch, 'A' as vendor, DAC, switch||DAC as unique_DAC1, min(cac) over (partition by switch, DAC, maxdiffcac) startrange, max(endcac) over (partition by switch, DAC, maxdiffcac) endrange

```
    from (
            select
                switch, DAC, cac, e, diffcac,
                max(diffcac) over (partition by switch, DAC order by cac RANGE BETWEEN UNBOUNDED PRECEDI
                nvl(lead(e) over (partition by switch, DAC order by cac), endcac) endcac

            from (
                     select
                         switch, DAC, cac, endcac, e,
                         sum(decode (diffcac, 1, 0, diffcac)) over (partition by switch, DAC order by ca
                     from (
                        select /*+DRIVING_SITE(a)*/
                                msc_id as switch,
                                lo_AREA_CODE as DAC,
                                START_se_AREA_CODE as cac,
                                END_se_AREA_CODE as endcac,
                                lag(END_se_AREA_CODE, 1,0) over (partition by msc_id, lo_AREA_CODE orde
                                (START_se_AREA_CODE - LAG(END_se_AREA_CODE, 1, 0) OVER (partition by ms

                        from L.DACSAABLE@OTHERDB, dates
                        where pull_date between date1 and date3
                         AND START_se_AREA_CODE<>END_se_AREA_CODE
                            and lo_AREA_CODE >= 64000
                     )
             )
     )
)
```

,

A2 as (select distinct DAC, switch, 'A' as vendor, min(cac) over (partition by switch, DAC, maxdiffcac) startrange, max(cac) over (partition by switch, DAC, maxdiffcac) endrange

from (select switch, DAC, cac, e, diffcac, max(diffcac) over (partition by switch, DAC order by cac RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) maxdiffcac

```
    from (

             select
                 switch,
                 DAC,
                 cac,
                 e,
                 sum(decode (diffcac, 1, 0, diffcac)) over (partition by switch, DAC order by cac RANGE

             from (
```

select */+DRIVING_SITE(a)/* msc_id as switch, lo_AREA_CODE as DAC, START_se_AREA_CODE as cac, lag(START_se_AREA_CODE, 1,0) over (partition by msc_id, lo_AREA_CODE order by START_se_AREA_CODE) E, (START_se_AREA_CODE - LAG(START_se_AREA_CODE, 1, 0) OVER (partition by msc_id, lo_AREA_CODE ORDER BY START_se_AREA_CODE)) diffcac

```
    from L.DACSTABLE@OTHERDB, dates
    where pull_date between date1 and date3
```

```
AND START_se_AREA_CODE = END_se_AREA_CODE
       and lo_AREA_CODE >= 64000
       )))),
```

step1 as (SELECT
distinct msc_id as switch1, trunc(pull_date) as dow1, loareacode as DAC1, msc_id||loareacode as uniqueid2
FROM MSCCM_WIREFIGUTRANDAC@OTHERDB, dates
where pull_date between date1 and date3 AND loareacode > = 64000), step2 as (
select switch1, dow1, DAC1, case when DAC is null then 'NULL' else 'FOUND' end as toss from step1 left join A on uniqueid2 = unique_DAC1),

A3 as (select switch1 as switch, DAC1 as DAC, 'A' as vendor, 0 as startrange, 0 as endrange, DAC1||'-0-0' as DACrange
from step2 wHERE TOSS = 'NULL'),

msc_join as (select switch, vendor, DAC, startrange, endrange, DAC||'-'||startrange||'-'||endrange as DACrange from e union all select switch, vendor, DAC, startrange, endrange, DAC||'-'||startrange||'-'||endrange as DACrange from a union all select switch, vendor, DAC, startrange, endrange, DAC||'-'||startrange||'-'||endrange as DACrange from a2 union all select switch, vendor, DAC, startrange, endrange, DAC||'-'||startrange||'-'||endrange as DACrange from a3

),

msc_range as (select distinct DACrange, switch, vendor, DAC,startrange,endrange,endrange-startrange+1 as alloc_cac_ct from msc_join union all select'NO_DAC' as DACrange,'UNKNOWN' as switch, 'UNKNOWN' as vendor, 0 as DAC,0 as startrange, 0 as endrange, 0 as alloc_cac_ct from dual

) SELECT * FROM MSC_RANGE
