SQL QUERY PERFORMANCE TUNING CASE STUDY

Tao Zuo

NPD Group

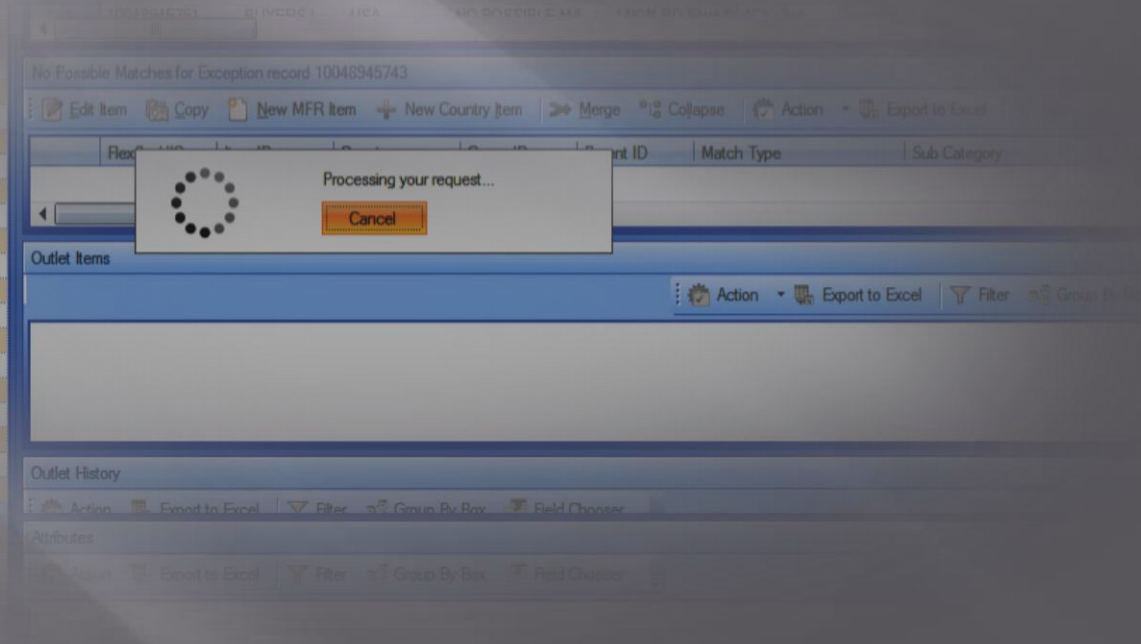
June, 2016

Introduction

- Oracle SQL tuning is a phenomenally complex subject.
- Assumption:
  Server/Instance/Object/optimizer-related level setting is optimal

Alarm Ring

- …process is running very long time (almost 45 mins)



Find the query

- V$SQL_MONITOR

```
SELECT inst_id,
       status,
       sid,
       session_serial#,
       username,
       module,
       action,
       service_name,
       client_identifier,
       client_info,
       program,
       cpu_time,
       queuing_time,
       elapsed_time, -- Elapsed time (in microseconds); updated as the statement executes
       sql_id,
       sql_text
  FROM gv$sql_monitor
 WHERE elapsed_time > 1000000 * 60 * 60
ORDER BY elapsed_time DESC;
```

Find the process

```
select s.username, s.sid, s.osuser, s.program,
s.machine, p.spid, s.status
from v$session s, v$process p
where s.paddr = p.addr
and s.sid in (922)
;
```

The query

- 92 lines,
- 4 layer subqueries
- 4 views

The query 1/7

```
SELECT e.poiid AS poiid, sopout101code AS sop_out101_code,
       sopout101name AS sop_out101_name, count11name AS count11,
       count11code AS count11_code, 0 AS matched_count11_code,
       out101code AS out101_code, out101name AS out101_name, subchan102_name,
       chan102_name, out101division AS out101_division,
       out101de8888mmmm AS out101_de8888mmmm,
       out101subde8888mmmm AS out101_sub_de8888mmmm,
       out101class AS out101_class, out101subclass AS out101_sub_class,
       uks AS uks, rfmcode AS rfm_code, out101br103 AS out101_br103,
       out101itemnumber AS out101_item_number,
       out101description AS out101_description, rfmcodetype AS rfm_code_type,
       manu4444tttrcodestatus AS manu4444tttr_code_status,
       out101descriptionppuslement AS out101_description_ppuslement,
       unitssold AS units_sold, totalvalue AS total_value,
       averageprice AS average_price, inventory AS inventory,
       out101br103match AS out101_br103_match,
       out101itemnumbermatch AS out101_item_number_match,
       out101descriptionmatch AS out101_description_match,
       manu4444tttrcodematch AS manu4444tttr_code_match,
       uksmatch AS uks_match, e.status AS status_code,
       (SELECT UPPER (description)
          FROM xyz_sopdatastatusvalues s
         WHERE s.FUNCTION = 'SopExceptions' AND s.status = e.status)
                                                  AS status,
       TO_CHAR (added, 'YYYY-MM-DD HH:MI AM') AS added,
       TO_CHAR (updated, 'YYYY-MM-DD HH:MI AM') AS updated,
       selected_categdog_code,
       CASE
          WHEN selected_categdog_code IS NOT NULL
             THEN categdogname
          ELSE NULL
       END AS selected_categdog_name,
       categdogcode AS actual_categdog_code,
       categdogname AS actual_categdog_name,
       categdogcodeset AS categdog_codeset, sub_categdog_code, categdog_code,
       categdog_name, br103code AS br103_code, br103name AS br103_name,
       percentofunits AS percent_of_units, percentofvalue AS percent_of_value,
```

The query 2/7

```
percentofinventory AS percent_of_inventory, NULL AS matched_item,
      ppmonth AS ppmonth, ppweek AS ppweek, YEAR AS YEAR, MONTH AS MONTH,
      week AS week, startdate AS start_date, enddate AS end_date,
      DECODE (cutoff_exp, 0, 'No', 'Yes') AS cutoff_exp,
      DECODE (hidden_exp, 0, 'No', 'Yes') AS hidden_exp,
      privatelabel AS private_label,
      origin_count11code AS origin_count11_code,
      (SELECT NAME
         FROM xyz_codes
        WHERE codeset = 'all_a_count11'
          AND code = origin_count11code) AS origin_count11_name,
      0 AS log_id, stone_weight AS carat, cut_grade AS cut,
      color_grade AS color, clarity, shape
  FROM (SELECT (SELECT code
                  FROM xyz_codes
                 WHERE bus466ssid = 471
                   AND codeset = 'djw_a_subcategdog'
                   AND NAME = categdogname) AS selected_categdog_code,
               e.*, cat.*, OUT.*
          FROM (SELECT e.*,
                       CASE
                         WHEN e.ppweek = 0
                           THEN (SELECT DISTINCT YEAR
                                            FROM xyz_calendar_weeks pp
                                           WHERE pp.calendar_id =
                                                        e.calendar_id
                                             AND pp.ppmonth = e.ppmonth)
                         ELSE (SELECT YEAR
                                 FROM xyz_calendar_weeks pp
                                WHERE pp.calendar_id = e.calendar_id
                                  AND pp.ppweek = e.ppweek
                                  AND pp.ppmonth = e.ppmonth)
                       END AS YEAR,
```

The query 3/7

```
CASE
                  WHEN e.ppweek = 0
                    THEN (SELECT DISTINCT TO_CHAR
                                 (TO_DATE (MONTH, 'MM'),
                                  'Month'
                                 ) AS MONTH
                            FROM xyz_calendar_weeks pp
                           WHERE pp.calendar_id =
                                    e.calendar_id
                             AND pp.ppmonth = e.ppmonth)
                  ELSE (SELECT TO_CHAR (TO_DATE (MONTH, 'MM'),
                                        'Month'
                                       ) AS MONTH
                          FROM xyz_calendar_weeks pp
                         WHERE pp.calendar_id = e.calendar_id
                           AND pp.ppweek = e.ppweek
                           AND pp.ppmonth = e.ppmonth)
                END AS MONTH,
                (SELECT 'Week ' || week
                  FROM xyz_calendar_weeks pp
                 WHERE pp.calendar_id = e.calendar_id
                   AND pp.ppweek = e.ppweek
                   AND pp.ppmonth = e.ppmonth) AS week,
                CASE
                  WHEN e.ppweek = 0
                    THEN NULL
                  ELSE (SELECT week_start_date
                          FROM xyz_calendar_weeks pp
                         WHERE pp.calendar_id =
                                  e.calendar_id
                           AND pp.ppweek = e.ppweek
                           AND pp.ppmonth = e.ppmonth)
                END AS startdate,
```

The query 4/7

```
CASE
                  WHEN e.ppweek = 0
                    THEN NULL
                  ELSE (SELECT week_end_date
                        FROM xyz_calendar_weeks pp
                        WHERE pp.calendar_id =
                                    e.calendar_id
                          AND pp.ppweek = e.ppweek
                          AND pp.ppmonth = e.ppmonth)
              END AS enddate,
              h.out101code, h.out101name,
              (SELECT count11_code
                FROM xyz_ret999ers
               WHERE ret999er_id = e.sopout101code) count11code,
              (SELECT NAME
                FROM xyz_codes
               WHERE code =
                     (SELECT count11_code
                       FROM xyz_ret999ers
                      WHERE ret999er_id =
                                e.sopout101code))
                                        count11name,
              h.br103codeset, h.percentofunits, percentofvalue,
              percentofinventory
         FROM (SELECT   lrowid, poiid, sopout101code, sopout101name,
                        out101division, out101de8888mmmm,
                        out101subde8888mmmm, out101class,
                        out101subclass, uks, out101br103,
                        out101itemnumber, out101description,
                        rfmcodetype, rfmcode, manu4444tttrcodestatus,
                        out101br103match, out101itemnumbermatch,
                        out101descriptionmatch,
                        manu4444tttrcodematch, uksmatch, itemtype,
                        status, added, updated, hidden_exp,
                        cutoff_exp, categdogcode, categdogname,
                        categdogcodeset, origin_count11code,
                        fld01code, fld01name, br103code, br103name,
```

The query 5/7

```
MIN (ppmonth) AS ppmonth,
                ROUND (MIN (inventory), 4) AS inventory,
                TRIM
                  (TO_CHAR ((CASE
                             WHEN SUM (unitssold) = 0
                               THEN 0
                             ELSE ROUND
                                    (  SUM (totalvalue)
                                     / SUM (unitssold),
                                     4
                                    )
                           END
                          ),
                          '$999999990.00'
                         )
                  ) AS averageprice,
                MIN (ppweek) AS ppweek,
                ROUND (SUM (unitssold), 4) AS unitssold,
                ROUND (SUM (totalvalue), 4) AS totalvalue,
                (SELECT calendar_id
                  FROM xyz_ret999ers
                 WHERE ret999er_id =
                          sopout101code)
                                AS calendar_id,
                MAX
                  (out101descriptionppuslement
                  ) AS out101descriptionppuslement,
                MAX (privatelabel) AS privatelabel
           FROM (SELECT *
                  FROM (SELECT a.*
                        FROM vw_djw_exceptions a
                       WHERE sopout101code = 253355
                         AND status NOT IN (91, -2)
                         AND status NOT IN
                                (99, 95, 94, 97)
                         AND status NOT IN (11, 12, 15)
                         AND hidden_exp = 0
```

The query 6/7

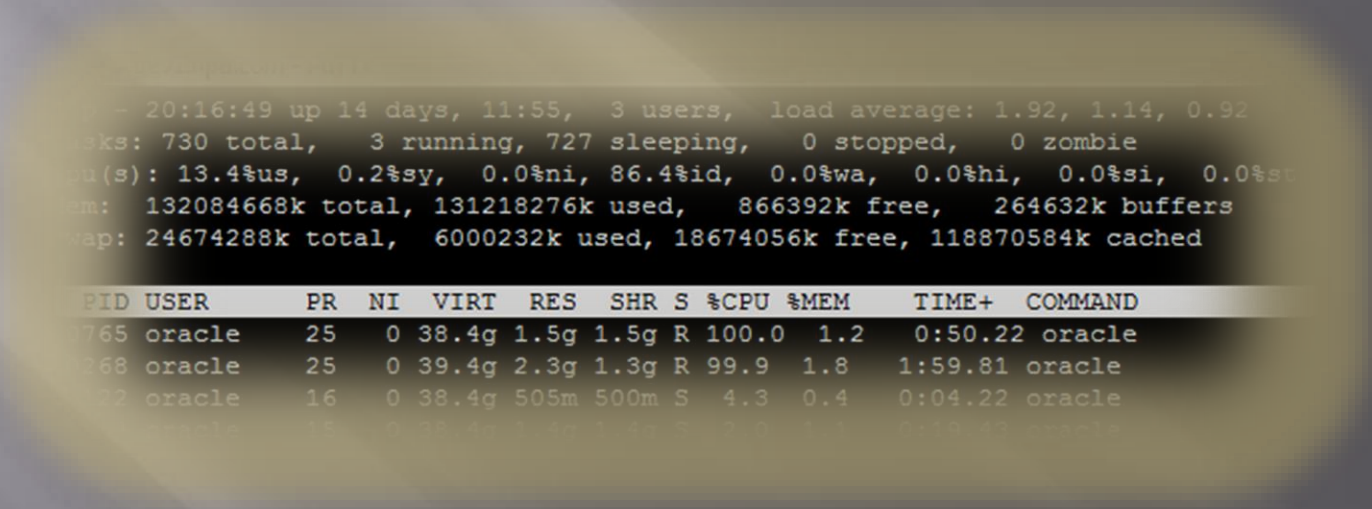
```
AND cutoff_exp = 0
                        AND ppmonth = 549))
            GROUP BY lrowid,
                  poiid,
                  sopout101code,
                  sopout101name,
                  out101division,
                  out101de8888mmmm,
                  out101subde8888mmmm,
                  out101class,
                  out101subclass,
                  uks,
                  out101br103,
                  out101itemnumber,
                  out101description,
                  rfmcodetype,
                  rfmcode,
                  manu4444tttrcodestatus,
                  out101br103match,
                  out101itemnumbermatch,
                  out101descriptionmatch,
                  manu4444tttrcodematch,
                  uksmatch,
                  itemtype,
                  status,
                  added,
                  updated,
                  hidden_exp,
                  cutoff_exp,
                  categdogcode,
                  categdogname,
                  categdogcodeset,
                  origin_count11code,
                  fld01code,
                  fld01name,
                  br103code,
                  br103name) e,
```

The query 7/7

```
(SELECT *
                  FROM vw_djw_sopout101s
                 WHERE sopout101code = 253355) h
           WHERE h.sopout101code(+) = e.sopout101code
             AND h.out101division(+) = NVL (e.out101division, '*')
             AND h.out101de8888mmmm(+) = NVL (e.out101de8888mmmm, '*')
             AND h.out101subde8888mmmm(+) =
                                 NVL (e.out101subde8888mmmm, '*')
             AND h.out101class(+) = NVL (e.out101class, '*')
             AND h.out101subclass(+) = NVL (e.out101subclass, '*')) e,
         vw_djw_categdoghierarchy_8_0 cat,
         vw_djw_out101hierlist_8_0 OUT
   WHERE 1 = 1 AND e.categdogcode = cat.code(+) AND e.out101code =
OUT.out101_code(+)) e,
     xyz_sopout101item_attributes a
WHERE e.sopout101code = a.sopout101(+) AND e.poiid = a.poiid(+);
```

Observation

Top utitily output:



Observation

OEM: Target: Databases->Database->Instance->Performance->Search Sessions->Session Details->Activity

OEM Tuning Adviser

SQL Text

select count(*) from (SELECT e.poiid AS poiid, ...

Select Recommendation

Original Explain Plan (Annotated)

Implement

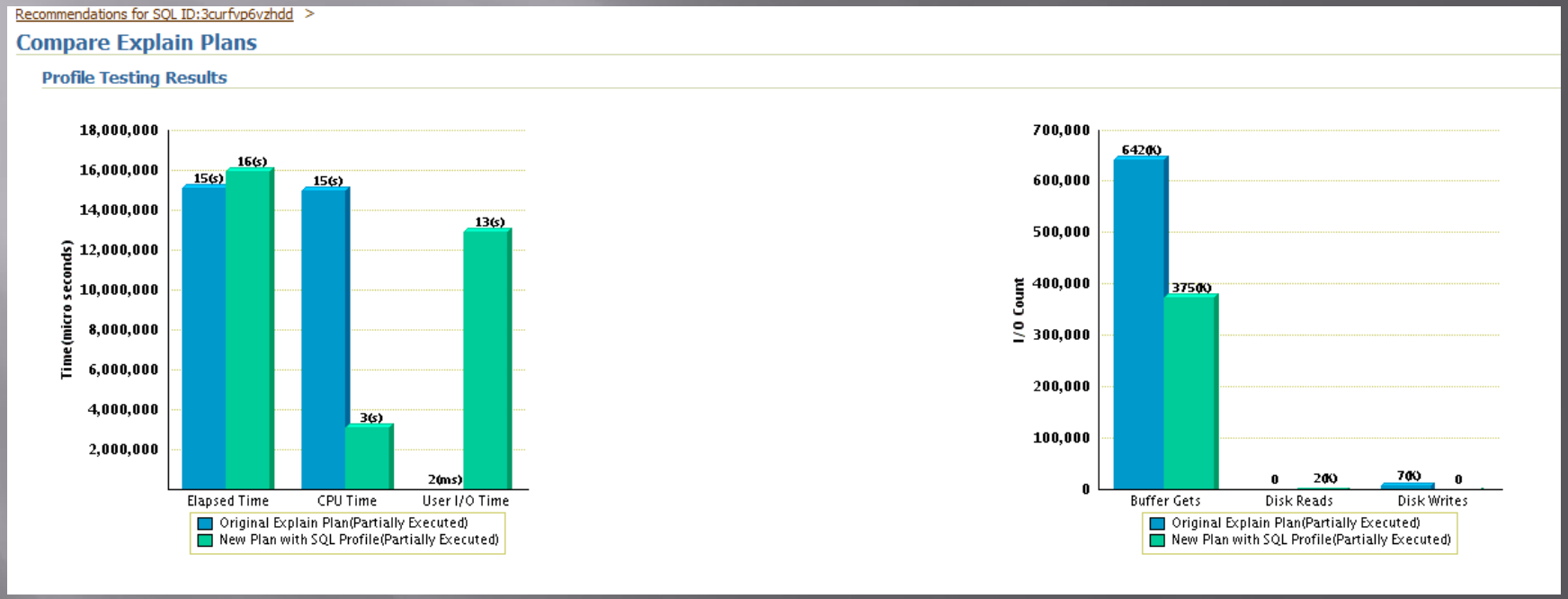
Select	Type	Findings	Recommendations	Rationale	Benefit (%)	Other Statistics	New Explain Plan	Compare Explain Plans
◉	SQL Profile	A potentially better execution plan was found for this statement.	SQL profile "SYS_SQLPROF_0153499bad0b0000" was created by the user for this recommendation and currently has status "ENABLED".		31.73			

OEM: Session Details->General tab>Current SQL(Application)->Schedule SQL Tuning

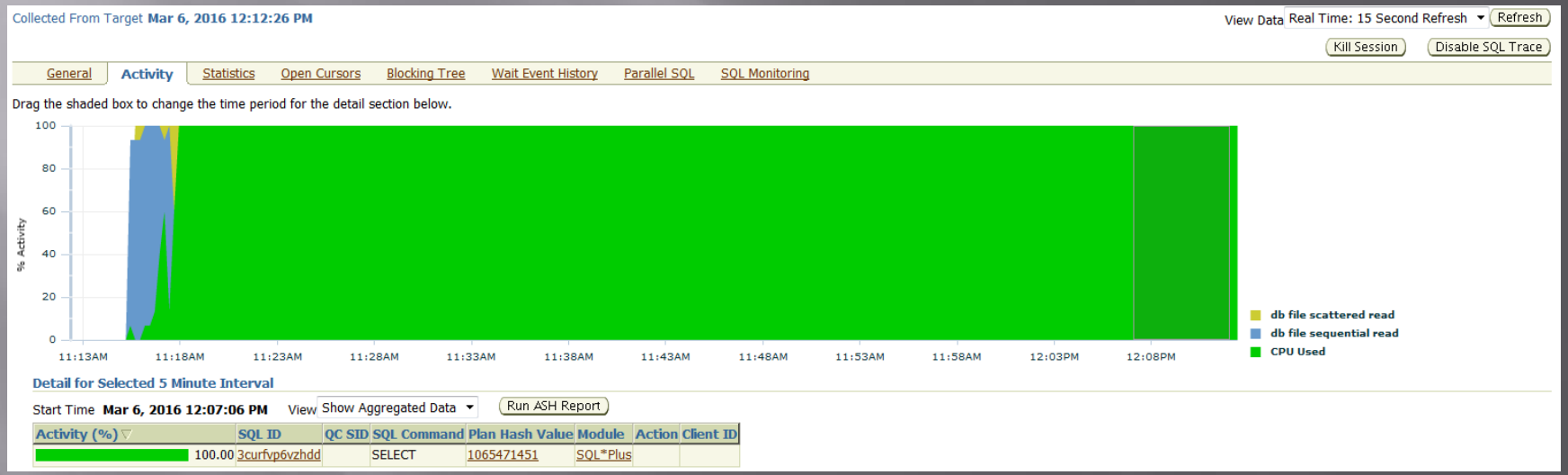
```
SQL> show parameter control_management_pack_access

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
control_management_pack_access       string      DIAGNOSTIC+TUNING
```

OEM Tuning Adviser

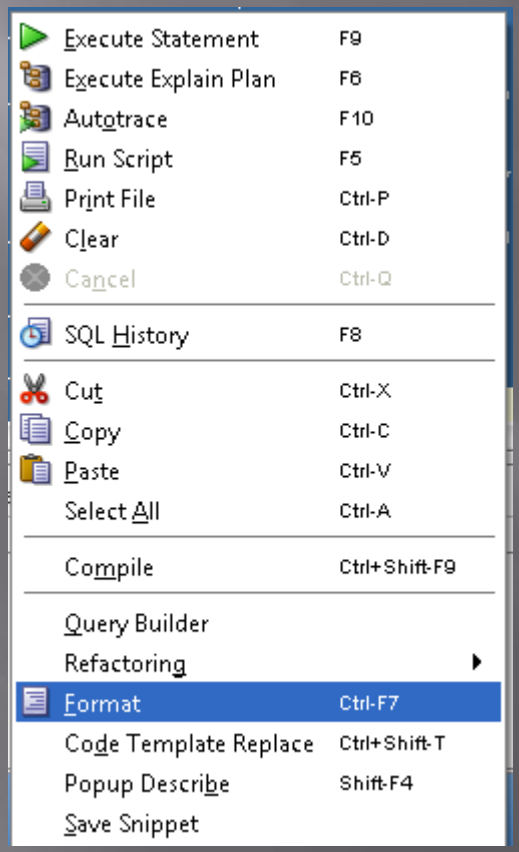


After accepting Running Adviser recommendation



Format the query

(Oracle SQL Developer)



The query after formated 1/11

```
SELECT e.poiid  AS poiid,
  sopout101code AS sop_out101_code,
  sopout101name AS sop_out101_name,
  count11name   AS count11,
  count11code   AS count11_code,
  0             AS matched_count11_code,
  out101code    AS out101_code,
  out101name    AS out101_name,
  subchan102_name,
  chan102_name,
  out101division              AS out101_division,
  out101de8888mmmm            AS out101_de8888mmmm,
  out101subde8888mmmm         AS out101_sub_de8888mmmm,
  out101class                 AS out101_class,
  out101subclass              AS out101_sub_class,
  uks                         AS uks,
  rfmcode                     AS rfm_code,
  out101br103                 AS out101_br103,
  out101itemnumber            AS out101_item_number,
  out101description           AS out101_description,
  rfmcodetype                 AS rfm_code_type,
  manu4444tttrcodestatus      AS manu4444tttr_code_status,
  out101descriptionppuslement AS out101_description_ppuslement,
  unitssold                   AS units_sold,
  totalvalue                  AS total_value,
  averageprice                AS average_price,
  inventory                   AS inventory,
```

The query after formated 2/11

```
out101br103match            AS out101_br103_match,
out101itemnumbermatch       AS out101_item_number_match,
out101descriptionmatch      AS out101_description_match,
manu4444tttrcodematch       AS manu4444tttr_code_match,
uksmatch                    AS uks_match,
e.status                    AS status_code,
(SELECT UPPER (description)
FROM xyz_sopdatastatusvalues s
WHERE s.FUNCTION = 'SopExceptions'
AND s.status     = e.status
)                                          AS status,
TO_CHAR (added, 'YYYY-MM-DD HH:MI AM')     AS added,
TO_CHAR (updated, 'YYYY-MM-DD HH:MI AM') AS updated,
selected_categdog_code,
CASE
  WHEN selected_categdog_code IS NOT NULL
  THEN categdogname
 SopSE NULL
END             AS selected_categdog_name,
categdogcode    AS actual_categdog_code,
categdogname    AS actual_categdog_name,
categdogcodeset AS categdog_codeset,
sub_categdog_code,
categdog_code,
categdog_name,
br103code                           AS br103_code,
br103name                           AS br103_name,
```

The query after formated 3/11

```
  percentofunits                          AS percent_of_units,
  percentofvalue                          AS percent_of_value,
  percentofinventory                      AS percent_of_inventory,
  NULL                                    AS matched_item,
  ppmonth                                 AS ppmonth,
  ppweek                                  AS ppweek,
  YEAR                                    AS YEAR,
  MONTH                                   AS MONTH,
  week                                    AS week,
  startdate                               AS start_date,
  enddate                                 AS end_date,
  DECODE (cutoff_exp, 0, 'No', 'Yes') AS cutoff_exp,
  DECODE (hidden_exp, 0, 'No', 'Yes') AS hidden_exp,
  privatelabel                            AS private_label,
  origin_count11code                      AS origin_count11_code,
  (SELECT NAME
  FROM xyz_codes
  WHERE codeset = 'all_a_count11'
  AND code      = origin_count11code
  )            AS origin_count11_name,
  0            AS log_id,
  stone_weight AS carat,
  cut_grade    AS cut,
  color_grade  AS color,
  clarity,
  shape
FROM
```

The query after formated 4/11

```
(SELECT
   (SELECT code
   FROM xyz_codes
   WHERE bus466ssid = 471
   AND codeset      = 'djw_a_subcategdog'
   AND NAME         = categdogname
   ) AS selected_categdog_code,
   e.*,
   cat.*,
   OUT.*
 FROM
   (SELECT e.*,
     CASE
       WHEN e.ppweek = 0
       THEN
         (SELECT DISTINCT YEAR
         FROM xyz_calendar_weeks pp
         WHERE pp.calendar_id = e.calendar_id
         AND pp.ppmonth       = e.ppmonth
         )
       ELSE
         (SELECT YEAR
         FROM xyz_calendar_weeks pp
         WHERE pp.calendar_id = e.calendar_id
         AND pp.ppweek        = e.ppweek
         AND pp.ppmonth       = e.ppmonth
         )
     END AS YEAR,
```

The query after formated 5/11

```
CASE
  WHEN e.ppweek = 0
  THEN
    (SELECT DISTINCT TO_CHAR (TO_DATE (MONTH, 'MM'), 'Month' ) AS MONTH
    FROM xyz_calendar_weeks pp
    WHERE pp.calendar_id = e.calendar_id
    AND pp.ppmonth       = e.ppmonth
    )
  ELSE
    (SELECT TO_CHAR (TO_DATE (MONTH, 'MM'), 'Month' ) AS MONTH
    FROM xyz_calendar_weeks pp
    WHERE pp.calendar_id = e.calendar_id
    AND pp.ppweek        = e.ppweek
    AND pp.ppmonth       = e.ppmonth
    )
END AS MONTH,
(SELECT 'Week '
  || week
FROM xyz_calendar_weeks pp
WHERE pp.calendar_id = e.calendar_id
AND pp.ppweek        = e.ppweek
AND pp.ppmonth       = e.ppmonth
) AS week,
```

The query after formated 6/11

```
CASE
    WHEN e.ppweek = 0
    THEN NULL
    ELSE
      (SELECT week_start_date
      FROM xyz_calendar_weeks pp
      WHERE pp.calendar_id = e.calendar_id
      AND pp.ppweek        = e.ppweek
      AND pp.ppmonth       = e.ppmonth
      )
  END AS startdate,
  CASE
    WHEN e.ppweek = 0
    THEN NULL
    ELSE
      (SELECT week_end_date
      FROM xyz_calendar_weeks pp
      WHERE pp.calendar_id = e.calendar_id
      AND pp.ppweek        = e.ppweek
      AND pp.ppmonth       = e.ppmonth
      )
  END AS enddate,
  h.out101code,
  h.out101name,
  (SELECT count11_code FROM xyz_ret999ers WHERE ret999er_id = e.sopout101code
  ) count11code,
```

The query after formated 7/11

```
(SELECT NAME
    FROM xyz_codes
    WHERE code =
      (SELECT count11_code FROM xyz_ret999ers WHERE ret999er_id = e.sopout101code
      )
    ) count11name,
    h.br103codeset,
    h.percentofunits,
    percentofvalue,
    percentofinventory
  FROM
    (SELECT lrowid,
      poiid,
      sopout101code,
      sopout101name,
      out101division,
      out101de8888mmmm,
      out101subde8888mmmm,
      out101class,
      out101subclass,
      uks,
      out101br103,
      out101itemnumber,
      out101description,
      rfmcodetype,
      rfmcode,
      manu4444tttrcodestatus,
```

The query after formated 8/11

```
out101br103match,
out101itemnumbermatch,
out101descriptionmatch,
manu4444tttrcodematch,
uksmatch,
itemtype,
status,
added,
updated,
hidden_exp,
cutoff_exp,
categdogcode,
categdogname,
categdogcodeset,
origin_count11code,
fld01code,
fld01name,
br103code,
br103name,
MIN (ppmonth)              AS ppmonth,
ROUND (MIN (inventory), 4) AS inventory,
TRIM (TO_CHAR ((
CASE
  WHEN SUM (unitssold) = 0
  THEN 0
  ELSE ROUND ( SUM (totalvalue) / SUM (unitssold), 4 )
END ), '$999999990.00' ) )  AS averageprice,
```

The query after formated 9/11

```
MIN (ppweek)                AS ppweek,
  ROUND (SUM (unitssold), 4)  AS unitssold,
  ROUND (SUM (totalvalue), 4) AS totalvalue,
  (SELECT calendar_id FROM xyz_ret999ers WHERE ret999er_id = sopout101code
  )                                  AS calendar_id,
  MAX (out101descriptionppuslement ) AS out101descriptionppuslement,
  MAX (privatelabel)                 AS privatelabel
FROM
  (SELECT *
  FROM
    (SELECT a.*
    FROM vw_djw_exceptions a
    WHERE sopout101code = 253355
    AND status NOT     IN (91, -2)
    AND status NOT     IN (99, 95, 94, 97)
    AND status NOT     IN (11, 12, 15)
    AND hidden_exp      = 0
    AND cutoff_exp      = 0
    AND ppmonth         = 549
    )
  )
GROUP BY lrowid,
  poiid,
  sopout101code,
  sopout101name,
  out101division,
  out101de8888mmmm,
```

The query after formated 10/11

```
out101subde8888mmmm,
out101class,
out101subclass,
uks,
out101br103,
out101itemnumber,
out101description,
rfmcodetype,
rfmcode,
manu4444tttrcodestatus,
out101br103match,
out101itemnumbermatch,
out101descriptionmatch,
manu4444tttrcodematch,
uksmatch,
itemtype,
status,
added,
updated,
hidden_exp,
cutoff_exp,
categdogcode,
categdogname,
categdogcodeset,
origin_count11code,
fld01code,
fld01name,
```

The query after formated 11/11*

```
        br103code,
        br103name
      ) e,
      (SELECT * FROM vw_djw_sopout101s WHERE sopout101code = 253355
      ) h
    WHERE h.sopout101code(+)     = e.sopout101code
    AND h.out101division(+)      = NVL (e.out101division, '*')
    AND h.out101de8888mmmm(+)    = NVL (e.out101de8888mmmm, '*')
    AND h.out101subde8888mmmm(+) = NVL (e.out101subde8888mmmm, '*')
    AND h.out101class(+)         = NVL (e.out101class, '*')
    AND h.out101subclass(+)      = NVL (e.out101subclass, '*')
    ) e,
    vw_djw_categdoghierarchy_8_0 cat,
    vw_djw_out101hierlist_8_0 OUT
  WHERE 1             = 1
  AND e.categdogcode = cat.code(+)
  AND e.out101code   = OUT.out101_code(+)
  ) e,
  xyz_sopout101item_attributes a
WHERE e.sopout101code = a.sopout101(+)
AND e.poiid           = a.poiid(+);
```

- Bad query block naming: Alias ‘e’ appears 3 times in different subquery block

The query after formated w. block Alias changed 11/11*

```
      br103code,
      br103name
    ) e3,
    (SELECT * FROM vw_djw_sopout101s WHERE sopout101code = 253355
    ) h
  WHERE h.sopout101code(+)     = e.sopout101code
  AND h.out101division(+)      = NVL (e.out101division, '*')
  AND h.out101de8888mmmm(+)    = NVL (e.out101de8888mmmm, '*')
  AND h.out101subde8888mmmm(+) = NVL (e.out101subde8888mmmm, '*')
  AND h.out101class(+)         = NVL (e.out101class, '*')
  AND h.out101subclass(+)      = NVL (e.out101subclass, '*')
  ) e2,
  vw_djw_categdoghierarchy_8_0 cat,
  vw_djw_out101hierlist_8_0 OUT
WHERE 1             = 1
AND e2.categdogcode = cat.code(+)
AND e2.out101code   = OUT.out101_code(+)
) e1,
xyz_sopout101item_attributes a1
WHERE e1.sopout101code = a1.sopout101(+)
AND e1.poiid           = a1.poiid(+);
```

- Change the 3 Alias ‘e’ for different subquery block to be e1,e2,e3.
- Change the alias of ‘a’ on the same layer as e3 to be a3

EXPLAIN PLAN

```
explain plan set statement_id='f' for
<QUERY>
;

select * from table(dbms_xplan.display);
```

EXPLAIN PLAN 1/6

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Plan hash value: 3464502560

------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                          | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     | Pstart| Pstop |
------------------------------------------------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                               | 94916 |   480M|       |   287K  (1)| 01:07:10 |       |       |
|   1 |  TABLE ACCESS BY GLOBAL INDEX ROWID      | XYZ_CODES                     |     1 |    25 |       |     3   (0)| 00:00:01 | ROWID | ROWID |
|*  2 |   INDEX UNIQUE SCAN                      | XYZ_CODES_UK01                |     1 |       |       |     2   (0)| 00:00:01 |       |       |
|   3 |  TABLE ACCESS BY GLOBAL INDEX ROWID      | XYZ_CODES                     |     1 |    25 |       |     3   (0)| 00:00:01 | ROWID | ROWID |
|*  4 |   INDEX UNIQUE SCAN                      | XYZ_CODES_UK01                |     1 |       |       |     2   (0)| 00:00:01 |       |       |
|   5 |    TABLE ACCESS BY INDEX ROWID           | XYZ_RET999ERS                 |     1 |    10 |       |     2   (0)| 00:00:01 |       |       |
|*  6 |     INDEX UNIQUE SCAN                    | XYZ_RET999ERS_PK              |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|   7 |  TABLE ACCESS BY INDEX ROWID             | XYZ_RET999ERS                 |     1 |    10 |       |     2   (0)| 00:00:01 |       |       |
|*  8 |   INDEX UNIQUE SCAN                      | XYZ_RET999ERS_PK              |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|   9 |  TABLE ACCESS BY GLOBAL INDEX ROWID      | XYZ_CODES                     |     1 |    25 |       |     3   (0)| 00:00:01 | ROWID | ROWID |
|* 10 |   INDEX UNIQUE SCAN                      | XYZ_CODES_UK01                |     1 |       |       |     2   (0)| 00:00:01 |       |       |
|  11 |  TABLE ACCESS BY INDEX ROWID             | XYZ_SOPDATASTATUSVALUES       |     1 |    36 |       |     1   (0)| 00:00:01 |       |       |
|* 12 |   INDEX UNIQUE SCAN                      | SYS_C002645765                |     1 |       |       |     0   (0)| 00:00:01 |       |       |
|  13 |  PARTITION LIST SINGLE                   |                               |     1 |    44 |       |     2   (0)| 00:00:01 |   KEY |   KEY |
|* 14 |   TABLE ACCESS BY LOCAL INDEX ROWID      | XYZ_CODES                     |     1 |    44 |       |     2   (0)| 00:00:01 |    59 |    59 |
|* 15 |    INDEX RANGE SCAN                      | XYZ_CODES_FK01                |     1 |       |       |     1   (0)| 00:00:01 |    59 |    59 |
|  16 |  PARTITION LIST SINGLE                   |                               |     1 |    28 |       |     2   (0)| 00:00:01 |   KEY |   KEY |
|  17 |   TABLE ACCESS BY LOCAL INDEX ROWID      | XYZ_CODES                     |     1 |    28 |       |     2   (0)| 00:00:01 |    59 |    59 |
|* 18 |    INDEX UNIQUE SCAN                     | XYZ_CODES_PK                  |     1 |       |       |     1   (0)| 00:00:01 |    59 |    59 |
|  19 |  PARTITION LIST SINGLE                   |                               |     1 |    27 |       |     2   (0)| 00:00:01 |   KEY |   KEY |
|  20 |   TABLE ACCESS BY LOCAL INDEX ROWID      | XYZ_CODES                     |     1 |    27 |       |     2   (0)| 00:00:01 |    59 |    59 |
|* 21 |    INDEX UNIQUE SCAN                     | XYZ_CODES_PK                  |     1 |       |       |     1   (0)| 00:00:01 |    59 |    59 |
|  22 |  TABLE ACCESS BY GLOBAL INDEX ROWID      | XYZ_CODES                     |     1 |    25 |       |     3   (0)| 00:00:01 | ROWID | ROWID |
|* 23 |   INDEX UNIQUE SCAN                      | XYZ_CODES_UK01                |     1 |       |       |     2   (0)| 00:00:01 |       |       |
|  24 |  SORT UNIQUE                             |                               |     3 |    33 |       |    30   (4)| 00:00:01 |       |       |
|* 25 |   TABLE ACCESS BY INDEX ROWID            | XYZ_CALENDAR_WEEKS            |     3 |    33 |       |    29   (0)| 00:00:01 |       |       |
|* 26 |    INDEX RANGE SCAN                      | XYZ_CALENDAR_WEEKS_FK01       |  4918 |       |       |     5   (0)| 00:00:01 |       |       |
|* 27 |    TABLE ACCESS BY INDEX ROWID           | XYZ_CALENDAR_WEEKS            |     1 |    15 |       |     2   (0)| 00:00:01 |       |       |
|* 28 |     INDEX UNIQUE SCAN                    | XYZ_CALENDAR_WEEKS_PK         |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|  29 |  SORT UNIQUE                             |                               |     3 |    30 |       |    30   (4)| 00:00:01 |       |       |
```

EXPLAIN PLAN 2/6

```
PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------
|* 30 |   TABLE ACCESS BY INDEX ROWID                | XYZ_CALENDAR_WEEKS           |     3 |    30 |       |    29   (0)| 00:00:01 |       |       |
|* 31 |    INDEX RANGE SCAN                          | XYZ_CALENDAR_WEEKS_FK01      |  4918 |       |       |     5   (0)| 00:00:01 |       |       |
|* 32 |    TABLE ACCESS BY INDEX ROWID               | XYZ_CALENDAR_WEEKS           |     1 |    14 |       |     2   (0)| 00:00:01 |       |       |
|* 33 |     INDEX UNIQUE SCAN                        | XYZ_CALENDAR_WEEKS_PK        |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|* 34 |  TABLE ACCESS BY INDEX ROWID                 | XYZ_CALENDAR_WEEKS           |     1 |    14 |       |     2   (0)| 00:00:01 |       |       |
|* 35 |   INDEX UNIQUE SCAN                          | XYZ_CALENDAR_WEEKS_PK        |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|* 36 |  TABLE ACCESS BY INDEX ROWID                 | XYZ_CALENDAR_WEEKS           |     1 |    19 |       |     2   (0)| 00:00:01 |       |       |
|* 37 |   INDEX UNIQUE SCAN                          | XYZ_CALENDAR_WEEKS_PK        |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|* 38 |  TABLE ACCESS BY INDEX ROWID                 | XYZ_CALENDAR_WEEKS           |     1 |    19 |       |     2   (0)| 00:00:01 |       |       |
|* 39 |   INDEX UNIQUE SCAN                          | XYZ_CALENDAR_WEEKS_PK        |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|  40 |  TABLE ACCESS BY INDEX ROWID                 | XYZ_RET999ERS                |     1 |    10 |       |     2   (0)| 00:00:01 |       |       |
|* 41 |   INDEX UNIQUE SCAN                          | XYZ_RET999ERS_PK             |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|* 42 |  TABLE ACCESS BY GLOBAL INDEX ROWID          | XYZ_CODES                    |     1 |    42 |       |     3   (0)| 00:00:01 | ROWID | ROWID |
|* 43 |   INDEX UNIQUE SCAN                          | XYZ_CODES_UK01               |     1 |       |       |     2   (0)| 00:00:01 |       |       |
|  44 |  TABLE ACCESS BY INDEX ROWID                 | XYZ_RET999ERS                |     1 |     8 |       |     2   (0)| 00:00:01 |       |       |
|* 45 |   INDEX UNIQUE SCAN                          | XYZ_RET999ERS_PK             |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|* 46 |  HASH JOIN RIGHT OUTER                       |                              | 94916 |   480M|       |   287K  (1)| 01:07:10 |       |       |
|  47 |   VIEW                                       | VW_DJW_OUT101HIERLIST_8_0    |     1 |   139 |       |     4   (0)| 00:00:01 |       |       |
|  48 |    NESTED LOOPS                              |                              |     1 |   149 |       |     4   (0)| 00:00:01 |       |       |
|  49 |     NESTED LOOPS                             |                              |     1 |   149 |       |     4   (0)| 00:00:01 |       |       |
|  50 |      NESTED LOOPS                            |                              |     1 |   121 |       |     3   (0)| 00:00:01 |       |       |
|  51 |       NESTED LOOPS                           |                              |     1 |    93 |       |     2   (0)| 00:00:01 |       |       |
|* 52 |        HASH JOIN                             |                              |     1 |    82 |       |     2   (0)| 00:00:01 |       |       |
|  53 |         PARTITION LIST SINGLE                |                              |     5 |   205 |       |     1   (0)| 00:00:01 |   KEY |   KEY |
|* 54 |          INDEX RANGE SCAN                    | XYZ_SUBCODES_PK              |     5 |   205 |       |     1   (0)| 00:00:01 |    59 |    59 |
|  55 |         PARTITION LIST SINGLE                |                              |   135 |  5535 |       |     1   (0)| 00:00:01 |   KEY |   KEY |
|* 56 |          INDEX RANGE SCAN                    | XYZ_SUBCODES_PK              |   135 |  5535 |       |     1   (0)| 00:00:01 |    59 |    59 |
|  57 |        PARTITION LIST SINGLE                 |                              |     1 |    11 |       |     0   (0)| 00:00:01 |   KEY |   KEY |
|* 58 |         INDEX UNIQUE SCAN                    | XYZ_CODES_PK                 |     1 |    11 |       |     0   (0)| 00:00:01 |    59 |    59 |
|  59 |       PARTITION LIST SINGLE                  |                              |     1 |    28 |       |     1   (0)| 00:00:01 |   KEY |   KEY |
|  60 |        TABLE ACCESS BY LOCAL INDEX ROWID     | XYZ_CODES                    |     1 |    28 |       |     1   (0)| 00:00:01 |    59 |    59 |
|* 61 |         INDEX UNIQUE SCAN                    | XYZ_CODES_PK                 |     1 |       |       |     0   (0)| 00:00:01 |    59 |    59 |
|  62 |      PARTITION LIST SINGLE                   |                              |     1 |       |       |     0   (0)| 00:00:01 |   KEY |   KEY |
|* 63 |       INDEX UNIQUE SCAN                      | XYZ_CODES_PK                 |     1 |       |       |     0   (0)| 00:00:01 |    59 |    59 |
|  64 |     TABLE ACCESS BY LOCAL INDEX ROWID        | XYZ_CODES                    |     1 |    28 |       |     1   (0)| 00:00:01 |    59 |    59 |
```

EXPLAIN PLAN 3/6

```
PLAN_TABLE_OUTPUT
-----------------------------------------------------------------------------------------------------------------------------------------------------------
|* 65 |   HASH JOIN RIGHT OUTER                        |                                | 94916 |   468M|       |   287K  (1)| 01:07:10 |       |       |
|  66 |    PARTITION LIST SINGLE                       |                                |    94 | 12502 |       |     5   (0)| 00:00:01 |   KEY |   KEY |
|  67 |     VIEW                                       |                                |    94 | 12502 |       |     5   (0)| 00:00:01 |       |       |
|  68 |      TABLE ACCESS BY LOCAL INDEX ROWID         | XYZ_BUSDE8888MMMMCLASSES       |    94 |  4700 |       |     5   (0)| 00:00:01 |    35 |    35 |
|* 69 |       INDEX RANGE SCAN                         | XYZ_BUSDE8888MMMMCLASSES_IE01  |     1 |       |       |     2   (0)| 00:00:01 |    35 |    35 |
|* 70 |    HASH JOIN RIGHT OUTER                       |                                | 94916 |   456M|       |   287K  (1)| 01:07:10 |       |       |
|  71 |     VIEW                                       |                                |     1 |    99 |       |    31   (4)| 00:00:01 |       |       |
|  72 |      HASH UNIQUE                               |                                |     1 |   156 |       |    31   (4)| 00:00:01 |       |       |
|  73 |       CONCATENATION                            |                                |       |       |       |             |          |       |       |
|  74 |        NESTED LOOPS                            |                                |     1 |   156 |       |    15   (0)| 00:00:01 |       |       |
|  75 |         NESTED LOOPS                           |                                |     1 |   156 |       |    15   (0)| 00:00:01 |       |       |
|* 76 |          HASH JOIN                             |                                |     1 |   112 |       |    14   (0)| 00:00:01 |       |       |
|* 77 |           HASH JOIN                            |                                |    32 |  2720 |       |    10   (0)| 00:00:01 |       |       |
|  78 |            PARTITION LIST SINGLE               |                                |    32 |  1312 |       |     1   (0)| 00:00:01 |   KEY |   KEY |
|* 79 |             INDEX RANGE SCAN                   | XYZ_SUBCODES_PK                |    32 |  1312 |       |     1   (0)| 00:00:01 |    59 |    59 |
|  80 |           PARTITION LIST SINGLE                |                                |  2809 |   120K|       |     9   (0)| 00:00:01 |   KEY |   KEY |
|* 81 |            TABLE ACCESS FULL                   | XYZ_CODES                      |  2809 |   120K|       |     9   (0)| 00:00:01 |    59 |    59 |
|  82 |          PARTITION LIST SINGLE                 |                                |    43 |  1161 |       |     4   (0)| 00:00:01 |   KEY |   KEY |
|  83 |           INLIST ITERATOR                      |                                |       |       |       |             |          |       |       |
|* 84 |            TABLE ACCESS BY LOCAL INDEX ROWID   | XYZ_CODES                      |    43 |  1161 |       |     4   (0)| 00:00:01 |    59 |    59 |
|* 85 |             INDEX RANGE SCAN                   | XYZ_CODES_FK01                 |     1 |       |       |     2   (0)| 00:00:01 |    59 |    59 |
|  86 |         PARTITION LIST SINGLE                  |                                |     1 |       |       |     0   (0)| 00:00:01 |   KEY |   KEY |
|* 87 |          INDEX UNIQUE SCAN                     | XYZ_CODES_PK                   |     1 |       |       |     0   (0)| 00:00:01 |    59 |    59 |
|* 88 |        TABLE ACCESS BY LOCAL INDEX ROWID       | XYZ_CODES                      |     1 |    44 |       |     1   (0)| 00:00:01 |    59 |    59 |
|  89 |       NESTED LOOPS                             |                                |     1 |   156 |       |    15   (0)| 00:00:01 |       |       |
|  90 |        NESTED LOOPS                            |                                |     1 |   156 |       |    15   (0)| 00:00:01 |       |       |
|* 91 |         HASH JOIN                              |                                |     1 |   112 |       |    14   (0)| 00:00:01 |       |       |
|* 92 |          HASH JOIN                             |                                |    32 |  2720 |       |    10   (0)| 00:00:01 |       |       |
|  93 |           PARTITION LIST SINGLE                |                                |    32 |  1312 |       |     1   (0)| 00:00:01 |   KEY |   KEY |
|* 94 |            INDEX RANGE SCAN                    | XYZ_SUBCODES_PK                |    32 |  1312 |       |     1   (0)| 00:00:01 |    59 |    59 |
|  95 |          PARTITION LIST SINGLE                 |                                |  2809 |   120K|       |     9   (0)| 00:00:01 |   KEY |   KEY |
|* 96 |           TABLE ACCESS FULL                    | XYZ_CODES                      |  2809 |   120K|       |     9   (0)| 00:00:01 |    59 |    59 |
|  97 |         PARTITION LIST SINGLE                  |                                |    43 |  1161 |       |     4   (0)| 00:00:01 |   KEY |   KEY |
|  98 |          INLIST ITERATOR                       |                                |       |       |       |             |          |       |       |
|* 99 |           TABLE ACCESS BY LOCAL INDEX ROWID    | XYZ_CODES                      |    43 |  1161 |       |     4   (0)| 00:00:01 |    59 |    59 |
```

EXPLAIN PLAN 4/6

```
PLAN_TABLE_OUTPUT
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
Plan hash value: 3464502560

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
| Id   | Operation                                | Name                          | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------
…
|*100 |              INDEX RANGE SCAN            | XYZ_CODES_FK01                |     1 |       |       |     2   (0)| 00:00:01 |    59 |    59 |
| 101 |           PARTITION LIST SINGLE          |                               |     1 |       |       |     0   (0)| 00:00:01 |   KEY |   KEY |
|*102 |            INDEX UNIQUE SCAN             | XYZ_CODES_PK                  |     1 |       |       |     0   (0)| 00:00:01 |    59 |    59 |
|*103 |          TABLE ACCESS BY LOCAL INDEX ROWID| XYZ_CODES                     |     1 |    44 |       |     1   (0)| 00:00:01 |    59 |    59 |
|*104 |      HASH JOIN RIGHT OUTER               |                               | 94916 |   447M|    91M|   287K  (1)| 01:07:09 |       |       |
| 105 |       PARTITION LIST ALL                 |                               |  2277K|    65M|       |  7013   (1)| 00:01:39 |     1 |  3191 |
| 106 |        TABLE ACCESS FULL                 | XYZ_SOPOUT101ITEM_ATTRIBUTES  |  2277K|    65M|       |  7013   (1)| 00:01:39 |     1 |  3191 |
| 107 |       VIEW                               |                               | 94916 |   444M|       |   257K  (1)| 01:00:10 |       |       |
| 108 |        TABLE ACCESS BY GLOBAL INDEX ROWID| XYZ_CODES                     |     1 |    25 |       |     3   (0)| 00:00:01 | ROWID | ROWID |
|*109 |         INDEX UNIQUE SCAN                | XYZ_CODES_UK01                |     1 |       |       |     2   (0)| 00:00:01 |       |       |
| 110 |       HASH GROUP BY                      |                               | 94916 |   248M|   296M|   257K  (1)| 01:00:10 |       |       |
| 111 |        VIEW                              |                               | 94916 |   248M|       |   223K  (1)| 00:52:15 |       |       |
| 112 |         NESTED LOOPS                     |                               | 94916 |    61M|       |   223K  (1)| 00:52:15 |       |       |
| 113 |          NESTED LOOPS                    |                               | 94916 |    61M|       |   223K  (1)| 00:52:15 |       |       |
|*114 |           HASH JOIN                      |                               | 94215 |    39M|    27M| 35414   (1)| 00:08:16 |       |       |
| 115 |            PART JOIN FILTER CREATE       | :BF0000                       | 94058 |    25M|       | 20176   (1)| 00:04:43 |       |       |
|*116 |             HASH JOIN                    |                               | 94058 |    25M|       | 20176   (1)| 00:04:43 |       |       |
| 117 |              PART JOIN FILTER CREATE     | :BF0001                       |     1 |    26 |       |     5   (0)| 00:00:01 |       |       |
| 118 |               PARTITION LIST SINGLE      |                               |     1 |    26 |       |     5   (0)| 00:00:01 |   KEY |   KEY |
|*119 |                TABLE ACCESS BY LOCAL INDEX ROWID| XYZ_SOPDATALOADSTATUS  |     1 |    26 |       |     5   (0)| 00:00:01 |    35 |    35 |
|*120 |                 INDEX RANGE SCAN         | XYZ_SOPDATALOADSTATUS_PK      |     1 |       |       |     1   (0)| 00:00:01 |    35 |    35 |
| 121 |              PARTITION RANGE SINGLE      |                               |  2818K|   706M|       | 20164   (1)| 00:04:43 |KEY(AP)|KEY(AP)|
|*122 |               TABLE ACCESS FULL          | XYZ_SOPUNMATCHEDDATA          |  2818K|   706M|       | 20164   (1)| 00:04:43 |   249 |   249 |
| 123 |            PARTITION LIST SINGLE         |                               |  1353K|   188M|       |  5372   (1)| 00:01:16 |KEY(AP)|KEY(AP)|
|*124 |             TABLE ACCESS FULL            | XYZ_SOPEXCEPTIONS             |  1353K|   188M|       |  5372   (1)| 00:01:16 |    35 |    35 |
|*125 |           INDEX UNIQUE SCAN              | XYZ_SOPOUT101ITEMS_UK01       |     1 |       |       |     1   (0)| 00:00:01 |       |       |
|*126 |          TABLE ACCESS BY GLOBAL INDEX ROWID| XYZ_SOPOUT101ITEMS          |     1 |   240 |       |     2   (0)| 00:00:01 | ROWID | ROWID |
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
   2 - access("CODE"=:B1)
   4 - access("CODE"= (SELECT "COUNT11_CODE" FROM "XYZ"."XYZ_RET999ERS" "XYZ_RET999ERS" WHERE "RET999ER_ID"=:B1))
   6 - access("RET999ER_ID"=:B1)
   8 - access("RET999ER_ID"=:B1)
```

EXPLAIN PLAN 5/6

```
10 - access("CODE"=:B1)
12 - access("S"."FUNCTION"='SopExceptions' AND "S"."STATUS"=:B1)
14 - filter("NAME"=:B1)
15 - access("CODESET"='djw_a_subcategdog')
18 - access("BUS466SSID"=471 AND "CODE"=:B1)
21 - access("BUS466SSID"=471 AND "CODE"=:B1)
23 - access("CODE"=:B1)
25 - filter("PP"."PPMONTH"=:B1)
26 - access("PP"."CALENDAR_ID"=:B1)
27 - filter("PP"."PPMONTH"=:B1)
28 - access("PP"."CALENDAR_ID"=:B1 AND "PP"."PPWEEK"=:B2)
30 - filter("PP"."PPMONTH"=:B1)
31 - access("PP"."CALENDAR_ID"=:B1)
32 - filter("PP"."PPMONTH"=:B1)
33 - access("PP"."CALENDAR_ID"=:B1 AND "PP"."PPWEEK"=:B2)
34 - filter("PP"."PPMONTH"=:B1)
35 - access("PP"."CALENDAR_ID"=:B1 AND "PP"."PPWEEK"=:B2)
36 - filter("PP"."PPMONTH"=:B1)
37 - access("PP"."CALENDAR_ID"=:B1 AND "PP"."PPWEEK"=:B2)
38 - filter("PP"."PPMONTH"=:B1)
39 - access("PP"."CALENDAR_ID"=:B1 AND "PP"."PPWEEK"=:B2)
41 - access("RET999ER_ID"=:B1)
42 - filter("CODESET"='all_a_count11')
43 - access("CODE"=:B1)
45 - access("RET999ER_ID"=:B1)
46 - access("H"."OUT101CODE"="OUT"."OUT101_CODE"(+))
52 - access("S1"."PARENTCODE"="S2"."CHILDCODE")
54 - access("S2"."BUS466SSID"=471 AND "S2"."SUBCODESET"='djw_a_chan102_subchan102')
56 - access("S1"."BUS466SSID"=471 AND "S1"."SUBCODESET"='djw_a_subchan102_OUT101')
58 - access("C1"."BUS466SSID"=471 AND "S1"."CHILDCODE"="C1"."CODE")
61 - access("C2"."BUS466SSID"=471 AND "S1"."PARENTCODE"="C2"."CODE")
63 - access("C3"."BUS466SSID"=471 AND "S2"."PARENTCODE"="C3"."CODE")
65 - access("H"."SOPOUT101CODE"(+)="E"."SOPOUT101CODE" AND "H"."OUT101DIVISION"(+)=NVL("E"."OUT101DIVISION",'*') AND
           "H"."OUT101DE8888MMMM"(+)=NVL("E"."OUT101DE8888MMMM",'*') AND "H"."OUT101SUBDE8888MMMM"(+)=NVL("E"."OUT101SUBDE8888MMMM",'*') AND
           "H"."OUT101CLASS"(+)=NVL("E"."OUT101CLASS",'*') AND "H"."OUT101SUBCLASS"(+)=NVL("E"."OUT101SUBCLASS",'*'))
69 - access("O"."SOPOUT101"=253355)
```

EXPLAIN PLAN 6/6

```
  70 - access("E"."CATGEDOGCODE"="CODE"(+))
  76 - access("C2"."CODE"="C"."CODE")
  77 - access("S1"."PARENTCODE"="C2"."CODE")
  79 - access("S1"."BUS466SSID"=471 AND "S1"."SUBCODESET"='djw_a_category_subcategdog')
  81 - filter("C2"."BUS466SSID"=471)
  84 - filter("C"."BUS466SSID"=471)
  85 - access("CODESET"='djw_a_categdog' OR "CODESET"='djw_a_subcategdog')
  87 - access("C1"."BUS466SSID"=471 AND "S1"."CHILDCODE"="C1"."CODE")
  88 - filter("C1"."CODESET"="C"."CODESET" OR "C2"."CODESET"="C"."CODESET")
  91 - access("C1"."CODE"="C"."CODE")
  92 - access("S1"."CHILDCODE"="C1"."CODE")
  94 - access("S1"."BUS466SSID"=471 AND "S1"."SUBCODESET"='djw_a_categdog_subcategdog')
  96 - filter("C1"."BUS466SSID"=471)
  99 - filter("C"."BUS466SSID"=471)
 100 - access("CODESET"='djw_a_categdog' OR "CODESET"='djw_a_subcategdog')
 102 - access("C2"."BUS466SSID"=471 AND "S1"."PARENTCODE"="C2"."CODE")
       filter(LNNVL("C2"."CODE"="C"."CODE"))
 103 - filter("C1"."CODESET"="C"."CODESET" OR "C2"."CODESET"="C"."CODESET")
 104 - access("E"."SOPOUT101CODE"="A"."SOPOUT101"(+) AND "E"."POIID"="A"."POIID"(+))
 109 - access("CODE"=:B1)
 114 - access("S"."BUS466SSID"="PE"."BUS466SSID" AND "PE"."POIID"="UMD"."POIID")
 116 - access("S"."PPMONTH"="UMD"."PPMONTH" AND "S"."PPWEEK"="UMD"."PPWEEK" AND "UMD"."LOADID"="S"."LOADID" AND
             "UMD"."SOPOUT101"="S"."SOPOUT101")
 119 - filter("S"."STATUS"=0 AND "S"."EXCEPTIONSTATUS"=0)
 120 - access("S"."PPMONTH"=549 AND "S"."SOPOUT101"=253355 AND "S"."BUS466SSID"=471)
       filter("S"."SOPOUT101"=253355)
 122 - filter("UMD"."SOPOUT101"=253355 AND "UMD"."PPMONTH"=549 AND ("UMD"."STATUS"=(-5) OR "UMD"."STATUS"=0))
 124 - filter("PE"."STATUS"<>91 AND "PE"."LOCKED"=0 AND "PE"."STATUS"<>11 AND "PE"."STATUS"<>12 AND "PE"."STATUS"<>15 AND
"PE"."STATUS"<>94 AND
             "PE"."STATUS"<>95 AND "PE"."STATUS"<>97 AND "PE"."STATUS"<>99 AND "PE"."STATUS"<>(-2))
 125 - access("PE"."POIID"="POI"."POIID")
 126 - filter("PE"."SOPOUT101"="POI"."SOPOUT101")
```

Refresh Statistics

- Make sure all statistics of the objects involved are up to date
  - DBA_TAB_MODIFICATIONS
- Refresh statistics
  - DBMS_STATS

Analysis

110	HASH GROUP BY		94916	248M	296M	257K	(1)	01:00:10			
111	VIEW		94916	248M		223K	(1)	00:52:15			
112	NESTED LOOPS		94916	61M		223K	(1)	00:52:15			
113	NESTED LOOPS		94916	61M		223K	(1)	00:52:15			
*114	HASH JOIN		94215	39M	27M	35414	(1)	00:08:16			
115	PART JOIN FILTER CREATE	:BF0000	94058	25M		20176	(1)	00:04:43			
*116	HASH JOIN		94058	25M		20176	(1)	00:04:43			
117	PART JOIN FILTER CREATE	:BF0001	1	26		5	(0)	00:00:01			
118	PARTITION LIST SINGLE		1	26		5	(0)	00:00:01	KEY	KEY	
*119	TABLE ACCESS BY LOCAL INDEX ROWID	XYZ_SOPDATALOADSTATUS	1	26		5	(0)	00:00:01	35	35	
*120	INDEX RANGE SCAN	XYZ_SOPDATALOADSTATUS_PK	1			1	(0)	00:00:01	35	35	
121	PARTITION RANGE SINGLE		2818K	706M		20164	(1)	00:04:43	KEY(AP)	KEY(AP)	
*122	TABLE ACCESS FULL	XYZ_SOPUNMATCHEDDATA	2818K	706M		20164	(1)	00:04:43	249	249	
123	PARTITION LIST SINGLE		1353K	188M		5372	(1)	00:01:16	KEY(AP)	KEY(AP)	
*124	TABLE ACCESS FULL	XYZ_SOPEXCEPTIONS	1353K	188M		5372	(1)	00:01:16	35	35	
*125	INDEX UNIQUE SCAN	XYZ_SOPOUT101ITEMS_UK01	1			1	(0)	00:00:01			
*126	TABLE ACCESS BY GLOBAL INDEX ROWID	XYZ_SOPOUT101ITEMS	1	240		2	(0)	00:00:01	ROWID	ROWID	

Analysis

```
BEGIN
  FOR V
    IN (SELECT VIEW_NAME, TEXT
         FROM DBA_VIEWS
        WHERE    OWNER = 'DICT_MGR'
             AND VIEW_NAME IN
                   (UPPER ('vw_djw_categdoghierarchy_8_0'),
                    UPPER ('vw_djw_out101hierlist_8_0'),
                    UPPER ('vw_djw_sopout101s'),
                    UPPER ('vw_djw_exceptions')))
  LOOP
    IF    INSTR (UPPER (V.TEXT), 'SOPDATALOADSTATUS') > 0
      AND INSTR (UPPER (V.TEXT), 'SOPUNMATCHEDDATA') > 0
      AND INSTR (UPPER (V.TEXT), 'SOPEXCEPTIONS') > 0
      AND INSTR (UPPER (V.TEXT), 'SOPOUT101ITEMS') > 0
    THEN
      DBMS_OUTPUT.PUT_LINE (V.VIEW_NAME);
    END IF;
  END LOOP;
END;
/

--------------
VW_DJW_EXCEPTIONS
```

Analysis

```
select count(*) from (
SELECT a.*
                                  FROM vw_djw_exceptions a
                                 WHERE sopout101code = 253355
                                   AND status NOT IN (91, -2)
                                   AND status NOT IN
                                             (99, 95, 94, 97)
                                   AND status NOT IN (11, 12, 15)
                                   AND hidden_exp = 0
                                   AND cutoff_exp = 0
                                   AND ppmonth = 549
) ;
-- 229,021 rows 23sec. Layer4 (bottom layer)
```

Analysis

```
SELECT lrowid,
        poiid,
        sopout101code,
        sopout101name,
        out101division,
        out101de8888mmmm,
        out101subde8888mmmm,
        out101class,
        out101subclass,
        uks,
        out101br103,
        out101itemnumber,
        out101description,
        rfmcodetype,
        rfmcode,
        manu4444tttrcodestatus,
        out101br103match,
        out101itemnumbermatch,
        out101descriptionmatch,
        manu4444tttrcodematch,
        uksmatch,
        itemtype,
        status,
        added,
        updated,
        hidden_exp,
        cutoff_exp,
        categdogcode,
        categdogname,
        categdogcodeset,
        origin_count11code,
        fld01code,
        fld01name,
        br103code,
        br103name,
        MIN (ppmonth)              AS ppmonth,
        ROUND (MIN (inventory), 4) AS inventory,
        TRIM (TO_CHAR ((
        CASE
          WHEN SUM (unitssold) = 0
          THEN 0
          ELSE ROUND ( SUM (totalvalue) / SUM (unitssold), 4 )
        END ), '$999999990.00' ) )  AS averageprice,
        MIN (ppweek)               AS ppweek,
        ROUND (SUM (unitssold), 4)  AS unitssold,
        ROUND (SUM (totalvalue), 4) AS totalvalue,
        (SELECT calendar_id FROM xyz_RET999ERs WHERE
ret999er_id = sopout101code
        )                                  AS calendar_id,
        MAX (out101descriptionppuslement ) AS
out101descriptionppuslement,
        MAX (privatelabel)                 AS privatelabel
      FROM (layer4)
```

-- 162869 rows 19sec

Analysis

```
SELECT e.*,
      CASE
        WHEN e.ppweek = 0
        THEN
          (SELECT DISTINCT YEAR
          FROM xyz_calendar_weeks pp
          WHERE pp.calendar_id = e.calendar_id
          AND pp.ppmonth       = e.ppmonth
          )
        ELSE
          (SELECT YEAR
          FROM xyz_calendar_weeks pp
          WHERE pp.calendar_id = e.calendar_id
          AND pp.ppweek        = e.ppweek
          AND pp.ppmonth       = e.ppmonth
          )
      END AS YEAR,
      CASE
        WHEN e.ppweek = 0
        THEN
          (SELECT DISTINCT TO_CHAR (TO_DATE (MONTH, 'MM'), 'Month' ) AS MONTH
          FROM xyz_calendar_weeks pp
          WHERE pp.calendar_id = e.calendar_id
          AND pp.ppmonth       = e.ppmonth
          )
        ELSE
          (SELECT TO_CHAR (TO_DATE (MONTH, 'MM'), 'Month' ) AS MONTH
          FROM xyz_calendar_weeks pp
          WHERE pp.calendar_id = e.calendar_id
          AND pp.ppweek        = e.ppweek
          AND pp.ppmonth       = e.ppmonth
          )
      END AS MONTH,
      (SELECT 'Week '
        || week
      FROM xyz_calendar_weeks pp
      WHERE pp.calendar_id = e.calendar_id
      AND pp.ppweek        = e.ppweek
      AND pp.ppmonth       = e.ppmonth
      ) AS week,
      CASE
        WHEN e.ppweek = 0
        THEN NULL
        ELSE
          (SELECT week_start_date
          FROM xyz_calendar_weeks pp
          WHERE pp.calendar_id = e.calendar_id
          AND pp.ppweek        = e.ppweek
          AND pp.ppmonth       = e.ppmonth
          )
      END AS startdate,
      CASE
        WHEN e.ppweek = 0
        THEN NULL
        ELSE
          (SELECT week_end_date
          FROM xyz_calendar_weeks pp
          WHERE pp.calendar_id = e.calendar_id
          AND pp.ppweek        = e.ppweek
          AND pp.ppmonth       = e.ppmonth
          )
      END AS enddate,
      h.out101code,
      h.out101name,
      (SELECT count11_code FROM xyz_ret999ers WHERE ret999er_id = e.sopout101code
      ) count11code,
      (SELECT NAME
      FROM xyz_codes
      WHERE code =
        (SELECT count11_code FROM xyz_ret999ers WHERE ret999er_id = e.sopout101code
        )
      ) count11name,
      h.br103codeset,
      h.percentofunits,
      percentofvalue,
      percentofinventory
    FROM (layer3)
```

– 162869 rows 6sec

Analysis

```
SELECT
    (SELECT code
    FROM xyz_codes
    WHERE bus466ssid = 471
    AND codeset      = 'djw_a_subcategdog'
    AND NAME         = categdogname
    ) AS selected_categdog_code,
    e.*,
    cat.*,
    OUT.*
  FROM (layer 2) – 162869 rows 7sec
```

Analysis

- top layer query -- this is the problematic layer.
- Execution plan: NESTED LOOPS

```
|  47 |   VIEW                                | VW_DJW_OUT101HIERLIST_8_0 |     1 |   139 |       |     4   (0)| 00:00:01 |       |       |
|  48 |    NESTED LOOPS                       |                           |     1 |   149 |       |     4   (0)| 00:00:01 |       |       |
|  49 |     NESTED LOOPS                      |                           |     1 |   149 |       |     4   (0)| 00:00:01 |       |       |
|  50 |      NESTED LOOPS                     |                           |     1 |   121 |       |     3   (0)| 00:00:01 |       |       |
|  51 |       NESTED LOOPS                    |                           |     1 |    93 |       |     2   (0)| 00:00:01 |       |       |
|* 52 |        HASH JOIN                      |                           |     1 |    82 |       |     2   (0)| 00:00:01 |       |       |
|  53 |         PARTITION LIST SINGLE         |                           |     5 |   205 |       |     1   (0)| 00:00:01 |   KEY |   KEY |
|* 54 |          INDEX RANGE SCAN             | XYZ_SUBCODES_PK           |     5 |   205 |       |     1   (0)| 00:00:01 |    59 |    59 |
|  55 |         PARTITION LIST SINGLE         |                           |   135 |  5535 |       |     1   (0)| 00:00:01 |   KEY |   KEY |
|* 56 |          INDEX RANGE SCAN             | XYZ_SUBCODES_PK           |   135 |  5535 |       |     1   (0)| 00:00:01 |    59 |    59 |
|  57 |        PARTITION LIST SINGLE          |                           |     1 |    11 |       |     0   (0)| 00:00:01 |   KEY |   KEY |
|* 58 |         INDEX UNIQUE SCAN             | XYZ_CODES_PK              |     1 |    11 |       |     0   (0)| 00:00:01 |    59 |    59 |
|  59 |       PARTITION LIST SINGLE           |                           |     1 |    28 |       |     1   (0)| 00:00:01 |   KEY |   KEY |
|  60 |        TABLE ACCESS BY LOCAL INDEX ROWID | XYZ_CODES                 |     1 |    28 |       |     1   (0)| 00:00:01 |    59 |    59 |
|* 61 |         INDEX UNIQUE SCAN             | XYZ_CODES_PK              |     1 |       |       |     0   (0)| 00:00:01 |    59 |    59 |
|  62 |      PARTITION LIST SINGLE            |                           |     1 |       |       |     0   (0)| 00:00:01 |   KEY |   KEY |
|* 63 |       INDEX UNIQUE SCAN               | XYZ_CODES_PK              |     1 |       |       |     0   (0)| 00:00:01 |    59 |    59 |
|  64 |     TABLE ACCESS BY LOCAL INDEX ROWID | XYZ_CODES                 |     1 |    28 |       |     1   (0)| 00:00:01 |    59 |    59 |
```

10046 trace

- ALTER SESSION SET TIMED_STATISTICS = TRUE;
- ALTER SESSION SET STATISTICS_LEVEL=TYPICAL;
- alter session set tracefile_identifier = 'SQL_badsql';
- alter session set events '10046 trace name context forever, level 12';
- <run query>
- alter session set events '10046 trace name context off';
- $ORACLE_BASE/diag/rdbms/<db>/<inst>/trace/<inst>_ora_xxxx_SQL_badsql.trc

Tkprof

- Tkprof /<inst>_ora_xxxx_SQL_badsql.trc <inst>_ora_xxxx_SQL_badsql.out

```
……
Line 1179:     162869     162869     162869   NESTED LOOPS OUTER (cr=396510507 pr=41355 pw=6690 time=1912696184 us cost=10384 size=192840 card=1607)
Line 1180:     162869     162869     162869    HASH JOIN RIGHT OUTER (cr=575968 pr=38937 pw=6690 time=68106257 us cost=9967 size=171949 card=1607)
Line 1181:        135        135        135     VIEW  VW_DJW_OUT101HIERLIST_8_0 (cr=16 pr=5 pw=0 time=50107 us cost=3 size=28 card=4)
Line 1182:        135        135        135      NESTED LOOPS  (cr=16 pr=5 pw=0 time=49903 us cost=3 size=476 card=4)
Line 1183:        135        135        135       NESTED LOOPS  (cr=12 pr=5 pw=0 time=48348 us cost=3 size=540 card=5)
Line 1184:        135        135        135        NESTED LOOPS  (cr=8 pr=5 pw=0 time=46650 us cost=3 size=485 card=5)
Line 1185:          5          5          5         NESTED LOOPS  (cr=5 pr=4 pw=0 time=31551 us cost=1 size=216 card=4)
Line 1186:          5          5          5          PARTITION LIST SINGLE PARTITION: KEY KEY (cr=3 pr=3 pw=0 time=21969 us cost=1 size=215 card=5)
Line 1187:          5          5          5           INDEX RANGE SCAN XYZ_SUBCODES_PK PARTITION: 59 59 (cr=3 pr=3 pw=0 time=21953 us cost=1 size=215 card=5)(object id 2493853)
Line 1188:          5          5          5          PARTITION LIST SINGLE PARTITION: KEY KEY (cr=2 pr=1 pw=0 time=9557 us cost=0 size=11 card=1)
Line 1189:          5          5          5           INDEX UNIQUE SCAN XYZ_CODES_PK PARTITION: 59 59 (cr=2 pr=1 pw=0 time=9531 us cost=0 size=11 card=1)(object id 2402829)
Line 1190:        135        135        135         PARTITION LIST SINGLE PARTITION: KEY KEY (cr=3 pr=1 pw=0 time=14958 us cost=1 size=43 card=1)
Line 1191:        135        135        135          TABLE ACCESS BY LOCAL INDEX ROWID XYZ_SUBCODES PARTITION: 59 59 (cr=3 pr=1 pw=0 time=14682 us cost=1 size=43 card=1)
Line 1192:        135        135        135           INDEX RANGE SCAN XYZ_SUBCODES_FK02 PARTITION: 59 59 (cr=2 pr=0 pw=0 time=44 us cost=0 size=0 card=1)(object id 2493907)
Line 1193:        135        135        135        PARTITION LIST SINGLE PARTITION: KEY KEY (cr=4 pr=0 pw=0 time=925 us cost=0 size=11 card=1)
Line 1194:        135        135        135         INDEX UNIQUE SCAN XYZ_CODES_PK PARTITION: 59 59 (cr=4 pr=0 pw=0 time=344 us cost=0 size=11 card=1)(object id 2402829)
Line 1195:        135        135        135       PARTITION LIST SINGLE PARTITION: KEY KEY (cr=4 pr=0 pw=0 time=882 us cost=0 size=11 card=1)
Line 1196:        135        135        135        INDEX UNIQUE SCAN XYZ_CODES_PK PARTITION: 59 59 (cr=4 pr=0 pw=0 time=359 us cost=0 size=11 card=1)(object id 2402829)
……
```

Top layer: time=1912696184 us : 32min.

2rd layer: time=68106257 us: 1min.

TKprof Interpretation (9i and above) (Doc ID 760786.1)

The query after formated w. block Alias changed, Hinted 1/11

```
SELECT /*+ USE_HASH(e1 a1) */
e1.poiid  AS poiid,
  sopout101code AS sop_out101_code,
  sopout101name AS sop_out101_name,
  count11name   AS count11,
  count11code   AS count11_code,
  0             AS matched_count11_code,
  out101code    AS out101_code,
  out101name    AS out101_name,
  subchan102_name,
  chan102_name,
  out101division              AS out101_division,
  out101de8888mmmm            AS out101_de8888mmmm,
  out101subde8888mmmm         AS out101_sub_de8888mmmm,
  out101class                 AS out101_class,
  out101subclass              AS out101_sub_class,
  uks                         AS uks,
  rfmcode                     AS rfm_code,
  out101br103                 AS out101_br103,
  out101itemnumber            AS out101_item_number,
  out101description           AS out101_description,
  rfmcodetype                 AS rfm_code_type,
  manu4444tttrcodestatus      AS manu4444tttr_code_status,
  out101descriptionppuslement AS out101_description_ppuslement,
  unitssold                   AS units_sold,
  totalvalue                  AS total_value,
  averageprice                AS average_price,
  inventory                   AS inventory,
```

EXPLAIN PLAN – After

```
-----------------------------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                     | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                          |     1 |   118 |       | 11132   (2)| 00:02:36 |       |       |
|   1 |  SORT AGGREGATE                          |                          |     1 |   118 |       |            |          |       |       |
|*  2 |   HASH JOIN OUTER                        |                          |  1498 |   172K|       | 11132   (2)| 00:02:36 |       |       |
|*  3 |    HASH JOIN RIGHT OUTER                 |                          |  1498 |   153K|       |  9811   (1)| 00:02:18 |       |       |
|   4 |     VIEW                                 | VW_DJW_OUT101HIERLIST_8_0 |     5 |    35 |       |     6  (17)| 00:00:01 |       |       |
```

• • •

Tkprof – After

- Tkprof – After

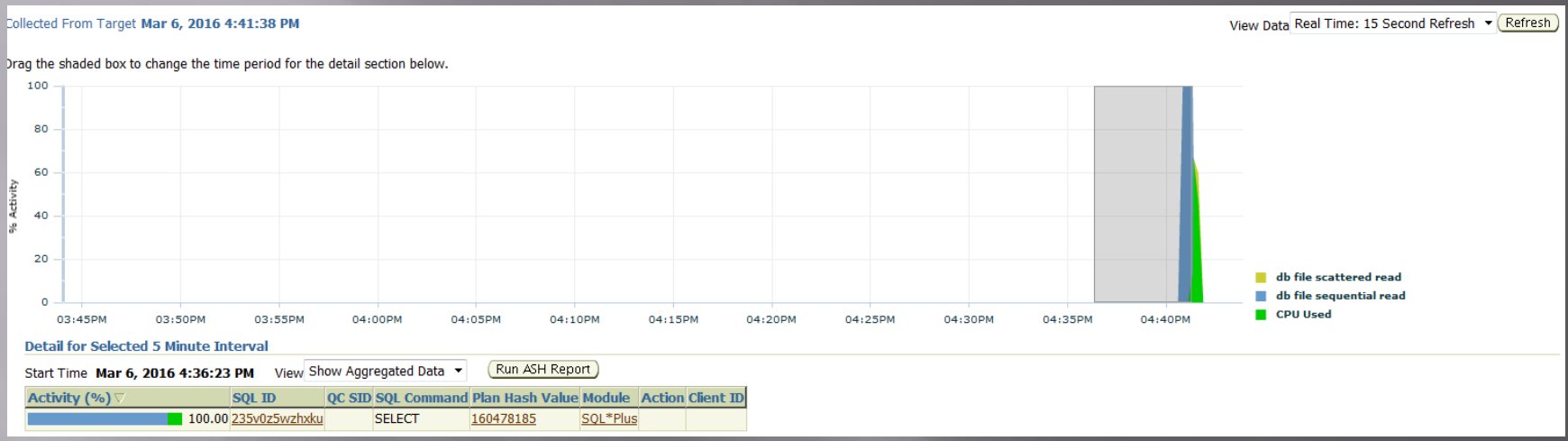
call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.51	0.58	0	48	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	2	16.18	31.17	16905	578400	0	1
total	4	16.69	31.76	16905	578448	0	1

```
Rows (1st) Rows (avg) Rows (max)  Row Source Operation
---------- ---------- ----------  ---------------------------------------------------
         1          1          1  SORT AGGREGATE (cr=578400 pr=16905 pw=6690 time=31172401 us)
    162869     162869     162869   HASH JOIN OUTER (cr=578400 pr=16905 pw=6690 time=31137179 us cost=10817 size=192840 card=1607)
    162869     162869     162869    HASH JOIN RIGHT OUTER (cr=575969 pr=16905 pw=6690 time=31278389 us cost=9967 size=171949 card=1607)
       135        135        135     VIEW  VW_DJW_OUT101HIERLIST_8_0 (cr=16 pr=0 pw=0 time=5063 us cost=3 size=28 card=4)
```

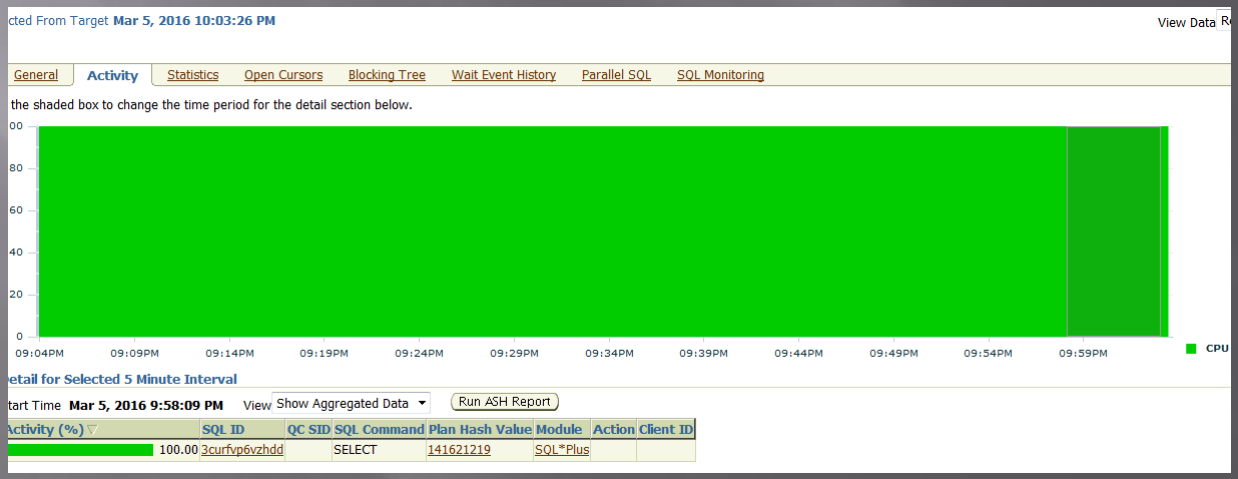
- Tkprof - before

```
Line 1179:     162869     162869     162869   NESTED LOOPS OUTER (cr=396510507 pr=41355 pw=6690 time=1912696104 us cost=10384 size=192840 card=1607)
Line 1180:     162869     162869     162869    HASH JOIN RIGHT OUTER (cr=575968 pr=38937 pw=6690 time=68106257 us cost=9967 size=171949 card=1607)
Line 1181:        135        135        135     VIEW  VW_DJW_OUT101HIERLIST_8_0 (cr=16 pr=5 pw=0 time=50107 us cost=3 size=28 card=4)
```

Observation – After



Before:



SQL PROFILE and HINT

- Hint works great, but I can't change the sql
- Use dbms_sqltune.import_sql_profile

SQL PROFILE and HINT

```
explain plan set statement_id='f' for
<QUERY hinted>
;
select * from
table(dbms_xplan.display(null,null,'ADVANCED'));
```

SQL PROFILE and HINT

<Run the query hinted>

```
SELECT * FROM
TABLE(DBMS_XPLAN.DISPLAY_CURSOR
(format => 'ADVANCED'));
```

SQL PROFILE and HINT

Outline Data

```
 /*+
     BEGIN_OUTLINE_DATA
     IGNORE_OPTIM_EMBEDDED_HINTS
     OPTIMIZER_FEATURES_ENABLE('11.2.0.3')
     DB_VERSION('11.2.0.4')
     OUTLINE_LEAF(@"SEL$19")
     OUTLINE_LEAF(@"SEL$43F09110")
…
     USE_HASH_AGGREGATION(@"SEL$43F09110")
     INDEX_RS_ASC(@"SEL$19" "XYZ_RET999ERS"@"SEL$19"
("XYZ_RET999ERS"."RET999ER_ID"))
     END_OUTLINE_DATA
  */
```

SQL PROFILE and HINT

```
DECLARE
    l_sql           clob;
    BEGIN
    l_sql := q'!select count(*) from (
SELECT e1.poiid  AS poiid,
…)
!';
   dbms_sqltune.import_sql_profile( sql_text => l_sql,
                            name => 'SQLPROFILE_01',
                            profile => sqlprof_attr(
q'! IGNORE_OPTIM_EMBEDDED_HINTS!',
q'! OPTIMIZER_FEATURES_ENABLE('11.2.0.3')!',
…
q'! USE_HASH_AGGREGATION(@"SEL$43F09110") !',
q'! INDEX_RS_ASC(@"SEL$19" "XYZ_RET999ERS"@"SEL$19" ("XYZ_RET999ERS"."RET999ER_ID"))
),
          force_match => true );
end;
/
```

SQL PROFILE and HINT

- Verify the execution plan

SQL> select * from table(dbms_xplan.display);

```
-----------------------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                    | Rows  | Bytes |TempSpc| Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                         |     1 |   118 |       | 11132   (2)| 00:02:36 |       |       |
|   1 |  SORT AGGREGATE                          |                         |     1 |   118 |       |            |          |       |       |
|*  2 |   HASH JOIN OUTER                        |                         |  1498 |   172K|       | 11132   (2)| 00:02:36 |       |       |
|*  3 |    HASH JOIN RIGHT OUTER                 |                         |  1498 |   153K|       |  9811   (1)| 00:02:18 |       |       |
|   4 |     VIEW                                 | VW_DJW_OUT101HIERLIST_8_0 |     5 |    35 |       |     6  (17)| 00:00:01 |       |       |
…
Note
-----
   - SQL profile "SQLPROFILE_01" used for this statement
```

Analysis Case 2

- Query has 107 lines
- Query calls several functions.

Query (1/3)

```
SELECT spr.sopt_edit_rule_id AS soptEditRuleId,
       spr.ppus_rule_id AS ppusRuleId,
       PPUS_PSTEDT_PROMOTION.CREATE_DISPLAY_FILTERS (spr.PPUS_rule_id,
                                                     spr.sopt_edit_rule_id)
         AS dataFilter,
       PPUS_PSTEDT_PROMOTION.CREATE_DISPLAY_CASES (spr.sopt_edit_rule_id)
         AS fieldFilter,
       PPUS_PSTEDT_PROMOTION.COMPILE_TRIGGER_STRING (spr.ppus_rule_id,
                                                     spr.sopt_edit_rule_id,
                                                     'D')
         AS processingFilter,
       DECODE (
         sfd.ppus_field_def_id,
         NULL, 'NA',
           UPPER (GET_SB_FIELD_NAME (sfd.ppus_field_def_id))
         || SUBSTR (sr.nexus_ppus_column, 11, LENGTH (sr.nexus_ppus_column)))
         AS fieldModified,
       spr.run_order_priority AS soptEditRunOrder,
       spr.field_priority AS fieldPriority,
       sfd.field_name AS fieldModified,
       spr.sopt_edit_rule_name AS soptEditRuleName,
       spr.sopt_edit_rule_desc AS soptEditRuleDesc,
       (SELECT type_name
          FROM LTE_TYPES
         WHERE type_id = action_type_id)
         AS actionType,
       action_type_id AS actionTypeId,
       (SELECT DISTINCT scd.name
          FROM LTE_PPUS_CODE_DEFS scd
         WHERE scd.code = spr.output_value)
         AS replaceWith,
       spr.output_value AS outputValue,
       sopt_edit_rule_type_id AS soptEditRuleTypeId,
       (SELECT type_name
          FROM LTE_TYPES
         WHERE type_id = sopt_edit_rule_type_id)
         AS soptEditRuleType,
```

Query 2 (2/3)

```
sfd.ppus_field_def_id AS outputppusFieldDefId,
spr.output_ppus_level_id AS outputPpusLevelId,
spc.sopt_edit_case_id AS casesoptEditCaseId,
spc.sopt_edit_rule_id AS casePostEditRuleId,
spc.dependent_ppus_level_id AS casePpusLevelId,
spc.dependent_Ppus_flag AS casePpusResultFlag,
(SELECT sfd.ppus_field_def_id AS ppusFieldDeffId
   FROM LTE_PPUS_FIELD_DEFS sfd, LTE_RPT_PPUSRESSION_LEVELS sl
  WHERE     sl.bus466ss = sfd.bus466ss
        AND sl.ppus_level_input = sfd.ppus_field_def_id
        AND ppus_level_id = spc.dependent_ppus_level_id)
   AS caseSoptEditCaseFieldDefId,
spc.added_date AS caseAddedDate,
spc.added_user AS caseAddedUser,
spc.updated_date AS caseUpdatedDate,
spc.updated_user AS caseUpdatedUser,
spr.added_date AS addedDate,
spr.added_user AS addedUser,
spr.updated_date AS updatedDate,
spr.updated_user AS updatedUser,
spfv.sopt_edit_rule_id AS fvSoptEditRuleId,
spfv.sequence_id AS fvSequenceId,
spfv.filter_field_num AS fvFilterFieldNum,
spfv.filter_field_value AS fvFilterFieldValue,
(SELECT c.name
   FROM XYZ_CODES C
  WHERE c.code = spfv.filter_field_value)
   AS fvFilterValueDescription,
spfv.exclude_flag AS fvExcludeFilterFlag,
spfv.added_date AS fvAddedDate,
spfv.added_user AS fvAddedUser,
spfv.updated_date AS fvUpdatedDate,
spfv.updated_user AS fvUpdatedUser,
spfd.filter_field_num AS filterFieldNumber,
spfd.ppus_field_def_id AS filterFieldId,
spfd.active_flag AS filterFieldActiveFlag,
```

Query 2 (3/3)

```
       DECODE (spfd.ppus_field_def_id,
               NULL, 'NA',
               UPPER (GET_SB_FIELD_NAME (spfd.supp_field_def_id)))
          AS filterFieldName,
       spfd.ppus_rule_id AS filterFieldPpusRuleId,
       spfd.added_date AS filterFieldAddedDate,
       spfd.updated_date AS filterFieldUpdatedDate,
       spfd.added_user AS filterFieldAddedUser,
       spfd.updated_user AS filterFieldUpdatedUser
  FROM LTE_RULESETS r,
       LTE_RPT_RULESET_VERSIONS rv,
       LTE_RPT_PPUSRESSION_RULES sr,
       LTE_RPT_PPUS_PSTEDT_RULES spr,
       LTE_RPT_ppusRESSION_LEVELS sl,
       LTE_PPUS_FIELD_DEFS sfd,
       (SELECT *
          FROM LTE_RPT_PPUS_PSTEDT_FIELD_DEFS
         WHERE active_flag = 1) spfd,
       LTE_RPT_PPUS_PSTEDT_CASES spc,
       LTE_RPT_PPUS_PSTEDT_FLT_VALUES spfv
 WHERE     r.ruleset_id = rv.ruleset_id
       AND rv.ruleset_version_id = sr.ppus_ruleset_version_id
       AND sr.ppus_rule_id = spr.ppus_rule_id
       AND spr.output_ppus_level_id = sl.ppus_level_id
       AND sl.bus466ss = r.bus466ss
       AND sl.ppus_level_input = sfd.ppus_field_def_id
       AND sfd.bus466ss = r.bus466ss
       AND spr.sopt_edit_rule_id = spc.sopt_edit_rule_id(+)
       AND spr.sopt_edit_rule_id = spfv.sopt_edit_rule_id(+)
       AND sr.ppus_rule_id = spfd.ppus_rule_id(+)
       AND spr.active_flag = 1
       AND spr.ppus_rule_id = 52574
       AND sfd.ppus_field_def_id = 246
ORDER BY spr.field_priority, spr.run_order_priority
```

EXPLAIN 1/2

```
Plan hash value: 1228438649

-----------------------------------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name                              | Rows  | Bytes | Cost (%CPU)| Time     | Pstart| Pstop |
-----------------------------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                                   |     9 |  3528 |   100   (1)| 00:00:02 |       |       |
|   1 |  TABLE ACCESS BY INDEX ROWID       | LTE_TYPES                         |     1 |    18 |     1   (0)| 00:00:01 |       |       |
|*  2 |   INDEX UNIQUE SCAN                | LTE_TYPES_PK                      |     1 |       |     0   (0)| 00:00:01 |       |       |
|   3 |  SORT UNIQUE                       |                                   |     1 |    21 |     3  (34)| 00:00:01 |       |       |
|   4 |   TABLE ACCESS BY INDEX ROWID      | LTE_PPUS_CODE_DEFS                |     1 |    21 |     2   (0)| 00:00:01 |       |       |
|*  5 |    INDEX RANGE SCAN                | LTE_PPUS_CODE_DEFS_PK             |     1 |       |     1   (0)| 00:00:01 |       |       |
|   6 |  TABLE ACCESS BY INDEX ROWID       | LTE_TYPES                         |     1 |    18 |     1   (0)| 00:00:01 |       |       |
|*  7 |   INDEX UNIQUE SCAN                | LTE_TYPES_PK                      |     1 |       |     0   (0)| 00:00:01 |       |       |
|   8 |  NESTED LOOPS                      |                                   |     1 |    22 |     3   (0)| 00:00:01 |       |       |
|   9 |   TABLE ACCESS BY INDEX ROWID      | LTE_RPT_PPUSRESSION_LEVELS        |     1 |    13 |     2   (0)| 00:00:01 |       |       |
|* 10 |    INDEX UNIQUE SCAN               | LTE_RPT_PPUSRESSION_LEVELS_PK     |     1 |       |     1   (0)| 00:00:01 |       |       |
|* 11 |   TABLE ACCESS BY INDEX ROWID      | LTE_PPUS_FIELD_DEFS               |     1 |     9 |     1   (0)| 00:00:01 |       |       |
|* 12 |    INDEX UNIQUE SCAN               | LTE_PPUS_FIELD_DEFS_PK            |     1 |       |     0   (0)| 00:00:01 |       |       |
|  13 |  TABLE ACCESS BY GLOBAL INDEX ROWID| XYZ_CODES                         |     1 |    25 |     3   (0)| 00:00:01 | ROWID | ROWID |
|* 14 |   INDEX UNIQUE SCAN                | XYZ_CODES_UK01                    |     1 |       |     2   (0)| 00:00:01 |       |       |
|  15 |  SORT ORDER BY                     |                                   |     9 |  3528 |   100   (1)| 00:00:02 |       |       |
|* 16 |   HASH JOIN OUTER                  |                                   |     9 |  3528 |    99   (0)| 00:00:02 |       |       |
|  17 |    NESTED LOOPS OUTER              |                                   |     3 |  1029 |    96   (0)| 00:00:02 |       |       |
|* 18 |     HASH JOIN OUTER                |                                   |     1 |   289 |    94   (0)| 00:00:02 |       |       |
|  19 |      NESTED LOOPS                  |                                   |     1 |   237 |    82   (0)| 00:00:02 |       |       |
|  20 |       NESTED LOOPS                 |                                   |     8 |   237 |    82   (0)| 00:00:02 |       |       |
|  21 |        NESTED LOOPS                |                                   |     8 |  1792 |    74   (0)| 00:00:02 |       |       |
|  22 |         NESTED LOOPS               |                                   |     1 |    82 |     4   (0)| 00:00:01 |       |       |
|  23 |          NESTED LOOPS              |                                   |     1 |    73 |     3   (0)| 00:00:01 |       |       |
|  24 |           NESTED LOOPS             |                                   |     1 |    61 |     2   (0)| 00:00:01 |       |       |
|  25 |            TABLE ACCESS BY INDEX ROWID| LTE_PPUS_FIELD_DEFS            |     1 |    25 |     1   (0)| 00:00:01 |       |       |
|* 26 |             INDEX UNIQUE SCAN      | LTE_PPUS_FIELD_DEFS_PK            |     1 |       |     0   (0)| 00:00:01 |       |       |
|  27 |            TABLE ACCESS BY INDEX ROWID| LTE_RPT_PPUSRESSION_RULES      |     1 |    36 |     1   (0)| 00:00:01 |       |       |
|* 28 |             INDEX UNIQUE SCAN      | LTE_RPT_PPUSRESSION_RULES_UK      |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 29 |           INDEX RANGE SCAN         | LTE_RPT_RULESET_VERSIONS_PK       |     1 |    12 |     1   (0)| 00:00:01 |       |       |
|* 30 |          TABLE ACCESS BY INDEX ROWID| LTE_RULESETS                      |     1 |     9 |     1   (0)| 00:00:01 |       |       |
```

EXPLAIN 2/2

```
|* 31 |           INDEX UNIQUE SCAN           | LTE_RULESETS_PK                |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 32 |         TABLE ACCESS FULL             | LTE_RPT_PPUS_PSTEDT_RULES      |     8 |  1136 |    70   (0)| 00:00:01 |       |       |
|* 33 |        INDEX UNIQUE SCAN              | LTE_RPT_PPUSRESSION_LEVELS_PK  |     1 |       |     0   (0)| 00:00:01 |       |       |
|* 34 |       TABLE ACCESS BY INDEX ROWID     | LTE_RPT_PPUSRESSION_LEVELS     |     1 |    13 |     1   (0)| 00:00:01 |       |       |
|  35 |      TABLE ACCESS FULL                | LTE_RPT_PPUS_PSTEDT_CASES      |  5781 |   293K|    12   (0)| 00:00:01 |       |       |
|  36 |     TABLE ACCESS BY INDEX ROWID       | LTE_RPT_PPUS_PSTEDT_FLT_VALUES |     3 |   162 |     2   (0)| 00:00:01 |       |       |
|* 37 |      INDEX RANGE SCAN                 | PPUS_PSTEDT_FLT_VALS_UNIQUE    |     3 |       |     1   (0)| 00:00:01 |       |       |
|* 38 |    TABLE ACCESS BY INDEX ROWID        | LTE_RPT_PPUS_PSTEDT_FIELD_DEFS |     3 |   147 |     3   (0)| 00:00:01 |       |       |
|* 39 |     INDEX RANGE SCAN                  | PPUS_PSTEDT_FIELD_DEFS_PK      |     3 |       |     2   (0)| 00:00:01 |       |       |
-----------------------------------------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("TYPE_ID"=:B1)
   5 - access("SCD"."CODE"=:B1)
   7 - access("TYPE_ID"=:B1)
  10 - access("PPUS_LEVEL_ID"=:B1)
  11 - filter("SL"."BUS466SS"="SFD"."BUS466SS")
  12 - access("SL"."PPUS_LEVEL_INPUT"="SFD"."PPUS_FIELD_DEF_ID")
  14 - access("C"."CODE"=:B1)
  16 - access("SR"."PPUS_RULE_ID"="LTE_RPT_PPUS_PSTEDT_FIELD_DEFS"."PPUS_RULE_ID"(+))
  18 - access("SPR"."SOPT_EDIT_RULE_ID"="SPC"."SOPT_EDIT_RULE_ID"(+))
  26 - access("SFD"."PPUS_FIELD_DEF_ID"=246)
  28 - access("SR"."PPUS_RULE_ID"=52574)
  29 - access("RV"."RULESET_VERSION_ID"="SR"."PPUS_RULESET_VERSION_ID")
  30 - filter("SFD"."BUS466SS"="R"."BUS466SS")
  31 - access("R"."RULESET_ID"="RV"."RULESET_ID")
  32 - filter("SPR"."PPUS_RULE_ID"=52574 AND "SPR"."ACTIVE_FLAG"=1)
  33 - access("SPR"."OUTPUT_PPUS_LEVEL_ID"="SL"."PPUS_LEVEL_ID")
  34 - filter("SL"."PPUS_LEVEL_INPUT"=246 AND "SL"."BUS466SS"="R"."BUS466SS")
  37 - access("SPR"."SOPT_EDIT_RULE_ID"="SPFV"."SOPT_EDIT_RULE_ID"(+))
  38 - filter("ACTIVE_FLAG"(+)=1)
  39 - access("LTE_RPT_PPUS_PSTEDT_FIELD_DEFS"."PPUS_RULE_ID"(+)=52574)

70 rows selected.
```

10046 Trace

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.02	0.02	0	3	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	36	0.17	0.17	0	1529	11404	35
total	38	0.20	0.20	0	1532	11404	35

10046 Trace

```
Rows (1st) Rows (avg) Rows (max)  Row Source Operation
---------- ---------- ----------  ---------------------------------------------------
         1          1          1  TABLE ACCESS BY INDEX ROWID LTE_TYPES (cr=2 pr=0 pw=0 time=26 us cost=1 size=18 card=1)
         1          1          1   INDEX UNIQUE SCAN LTE_TYPES_PK (cr=1 pr=0 pw=0 time=14 us cost=0 size=0 card=1)(object id 14779233)
         0          0          0  SORT UNIQUE (cr=0 pr=0 pw=0 time=15 us cost=3 size=21 card=1)
         0          0          0   TABLE ACCESS BY INDEX ROWID LTE_PPUS_CODE_DEFS (cr=0 pr=0 pw=0 time=6 us cost=2 size=21 card=1)
         0          0          0    INDEX RANGE SCAN LTE_PPUS_CODE_DEFS_PK (cr=0 pr=0 pw=0 time=2 us cost=1 size=0 card=1)(object id 15000131)
         1          1          1  TABLE ACCESS BY INDEX ROWID LTE_TYPES (cr=2 pr=0 pw=0 time=10 us cost=1 size=18 card=1)
         1          1          1   INDEX UNIQUE SCAN LTE_TYPES_PK (cr=1 pr=0 pw=0 time=4 us cost=0 size=0 card=1)(object id 14779233)
         2          2          2  NESTED LOOPS  (cr=9 pr=0 pw=0 time=66 us cost=3 size=22 card=1)
         2          2          2   TABLE ACCESS BY INDEX ROWID LTE_RPT_PPUSRESSION_LEVELS (cr=5 pr=0 pw=0 time=31 us cost=2 size=13 card=1)
         2          2          2    INDEX UNIQUE SCAN LTE_RPT_PPUSRESSION_LEVELS_PK (cr=3 pr=0 pw=0 time=17 us cost=1 size=0 card=1)(object id 15261028)
         2          2          2   TABLE ACCESS BY INDEX ROWID LTE_PPUS_FIELD_DEFS (cr=4 pr=0 pw=0 time=20 us cost=1 size=9 card=1)
         2          2          2    INDEX UNIQUE SCAN LTE_PPUS_FIELD_DEFS_PK (cr=2 pr=0 pw=0 time=7 us cost=0 size=0 card=1)(object id 14779217)
         3          3          3  TABLE ACCESS BY GLOBAL INDEX ROWID XYZ_CODES PARTITION: ROW LOCATION ROW LOCATION (cr=11 pr=0 pw=0 time=90 us cost=3 size=25 card=1)
         3          3          3   INDEX UNIQUE SCAN XYZ_CODES_UK01 (cr=8 pr=0 pw=0 time=55 us cost=2 size=0 card=1)(object id 15278521)
        35         35         35  SORT ORDER BY (cr=1277054 pr=3 pw=0 time=143806498 us cost=98 size=3096 card=8)
        35         35         35   HASH JOIN OUTER (cr=268 pr=0 pw=0 time=7117 us cost=97 size=3096 card=8)
         7          7          7    NESTED LOOPS OUTER (cr=265 pr=0 pw=0 time=4533 us cost=94 size=1020 card=3)
         5          5          5     HASH JOIN OUTER (cr=254 pr=0 pw=0 time=6370 us cost=92 size=288 card=1)
         5          5          5      NESTED LOOPS  (cr=228 pr=0 pw=0 time=1379 us cost=82 size=236 card=1)
        23         23         23       NESTED LOOPS  (cr=205 pr=0 pw=0 time=1728 us cost=82 size=236 card=8)
        23         23         23        NESTED LOOPS  (cr=196 pr=0 pw=0 time=803 us cost=74 size=1784 card=8)
         1          1          1         NESTED LOOPS  (cr=9 pr=0 pw=0 time=165 us cost=4 size=81 card=1)
         1          1          1          NESTED LOOPS  (cr=6 pr=0 pw=0 time=114 us cost=3 size=72 card=1)
         1          1          1           NESTED LOOPS  (cr=5 pr=0 pw=0 time=87 us cost=2 size=61 card=1)
         1          1          1            TABLE ACCESS BY INDEX ROWID LTE_PPUS_FIELD_DEFS (cr=2 pr=0 pw=0 time=41 us cost=1 size=25 card=1)
         1          1          1             INDEX UNIQUE SCAN LTE_PPUS_FIELD_DEFS_PK (cr=1 pr=0 pw=0 time=19 us cost=0 size=0 card=1)(object id 14779217)
         1          1          1            TABLE ACCESS BY INDEX ROWID LTE_RPT_PPUSRESSION_RULES (cr=3 pr=0 pw=0 time=40 us cost=1 size=36 card=1)
         1          1          1             INDEX UNIQUE SCAN LTE_RPT_PPUSRESSION_RULES_UK (cr=2 pr=0 pw=0 time=28 us cost=0 size=0 card=1)(object id 15546053)
         1          1          1           INDEX RANGE SCAN LTE_RPT_RULESET_VERSIONS_PK (cr=1 pr=0 pw=0 time=20 us cost=1 size=11 card=1)(object id 15260996)
         1          1          1          TABLE ACCESS BY INDEX ROWID LTE_RULESETS (cr=3 pr=0 pw=0 time=43 us cost=1 size=9 card=1)
         1          1          1           INDEX UNIQUE SCAN LTE_RULESETS_PK (cr=2 pr=0 pw=0 time=27 us cost=0 size=0 card=1)(object id 14779176)
        23         23         23         TABLE ACCESS FULL LTE_RPT_PPUS_PSTEDT_RULES (cr=187 pr=0 pw=0 time=627 us cost=70 size=1136 card=8)
        23         23         23        INDEX UNIQUE SCAN LTE_RPT_PPUSRESSION_LEVELS_PK (cr=9 pr=0 pw=0 time=576 us cost=0 size=0 card=1)(object id 15261028)
         5          5          5       TABLE ACCESS BY INDEX ROWID LTE_RPT_PPUSRESSION_LEVELS (cr=23 pr=0 pw=0 time=62 us cost=1 size=13 card=1)
      5266       5266       5266      TABLE ACCESS FULL LTE_RPT_PPUS_PSTEDT_CASES (cr=26 pr=0 pw=0 time=5546 us cost=10 size=266968 card=5134)
         6          6          6     TABLE ACCESS BY INDEX ROWID LTE_RPT_PPUS_PSTEDT_FLT_VALUES (cr=11 pr=0 pw=0 time=1459 us cost=2 size=156 card=3)
         6          6          6      INDEX RANGE SCAN PPUS_PSTEDT_FLT_VALS_UNIQUE (cr=7 pr=0 pw=0 time=403 us cost=1 size=0 card=3)(object id 17373910)
         5          5          5    TABLE ACCESS BY INDEX ROWID LTE_RPT_PPUS_PSTEDT_FIELD_DEFS (cr=3 pr=0 pw=0 time=872 us cost=3 size=141 card=3)
         5          5          5     INDEX RANGE SCAN PPUS_PSTEDT_FIELD_DEFS_PK (cr=2 pr=0 pw=0 time=371 us cost=2 size=0 card=3)(object id 17373906)
```

10046 Trace

- Query plan doesn't show any problem
- 10046 trace said it returns in 0.2 sec
- Where is the time spend?
- 10046 trace has more stories to tell
  - Many spin off calls from the query

10046 Trace

OVERALL TOTALS FOR ALL NON-RECURSIVE STATEMENTS

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.02	0.02	0	3	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	36	0.17	0.17	0	1529	11404	35
total	38	0.20	0.20	0	1532	11404	35

OVERALL TOTALS FOR ALL RECURSIVE STATEMENTS

call	count	cpu	elapsed	disk	query	current	rows
Parse	272	0.03	0.03	0	0	8	0
Execute	4147	0.36	0.37	0	1	5	1
Fetch	4313	143.19	143.23	3	1275525	0	3655
total	8732	143.59	143.63	3	1275526	13	3656

10046 Trace

```
$grep total DXYZ1_ora_16499_SQL_badsql.out
total       38      0.20       0.20          0       1532      11404          35
total      315      0.00       0.00          0        210          0          90
total      181      0.00       0.00          0        630          0          90
total      181      0.01       0.01          0        900          2         120
total      241      0.01       0.01          0        240          0         120
total      341      0.02       0.02          0       1870          0         170
total      241      0.00       0.00          0        240          0         120
total      451      0.01       0.01          0        630          0         150
total      301      0.00       0.01          0        390          0          90
total      141      0.03       0.02          0       2160          0          50
total       71      0.28       0.30          0       8785          0          35
total     1681      0.04       0.05          0       2520          0         840
total     1681      0.05       0.04          0       2520          0         840
total     1681      0.05       0.05          0       2730          0         350
total      701    142.97     143.00          0    1250690          0           0
total      105      0.00       0.00          0         70          0          35
total      105      0.00       0.00          0        385          0          35
total       21      0.00       0.00          0         14          0           7
total       21      0.00       0.00          0         77          0           7
total       21      0.00       0.00          0         14          0           7
total       21      0.00       0.00          0         77          0           7
total       21      0.00       0.00          0         14          0           7
total       21      0.00       0.00          0         77          0           7
total       21      0.00       0.00          0         14          0           7
total       21      0.00       0.00          0         77          0           7
total       21      0.00       0.00          0         14          0           7
total       21      0.00       0.00          0         77          0           7
total       38      0.20       0.20          0       1532      11404          35
total     8732    143.59     143.63          3    1275526         13        3656
```

10046 Trace

```
SELECT DISTINCT 1
FROM
 (SELECT TO_CHAR (OCC.CODE) AS CODE, OCC.NAME FROM
  LTE_PPUS_FIELD_DEFS FD, LTE_BUS_DATASRC_DATAELEMENTS DD,
  LTE_CODESET_MAPS CM, (SELECT SCD.CODE, SCD.CODESET, C.NAME FROM
  LTE_PPUS_CODE_DEFS SCD, XYZ_CODES C WHERE SCD.CODE = C.CODE AND
  ( C.NO_DELETE_FLAG = 1 AND LOWER (C.NAME) LIKE '%other%') UNION SELECT
  SCD.CODE, SCD.CODESET, CI.PARENT_NAME||'#'||CI.MODELNUMBER FROM
  LTE_PPUS_CODE_DEFS SCD, XYZ_CODE_ITEMNUMBERS CI WHERE SCD.CODE =
   CI.CODE AND ( CI.NO_DELETE_FLAG = 1 AND LOWER (CI.MODELNUMBER) LIKE
  '%other%' AND LOWER(CI.PARENT_NAME) LIKE '%other%')) OCC WHERE FD.DIM_LEVEL
  IN (SELECT DIMENSIONVIEW_LEVEL_ID FROM XYZ_DIMENSIONVIEW_LEVELS WHERE
  DIMENSIONVIEW = :B1 ) AND FD.INPUT_DATA_ELEMENT_ID = DD.BUSDS_ELEMENT_ID
  AND DD.CODESET_MAP_ID = CM.CODESET_MAP_ID AND CM.CODESET = OCC.CODESET
  UNION SELECT TO_CHAR (OCC.CODE) AS CODE, OCC.NAME FROM
  LTE_PPUS_FIELD_DEFS FD, LTE_BUS_DATASRC_DATAELEMENTS DD,
  LTE_CODESET_MAPS CM, XYZ_CODESETS CC, XYZ_DERIVEDCODESETS
  DC, (SELECT CV.CODE, C.CODESET, C.NAME FROM XYZ_CODES C,
  XYZ_CODEVIEWS CV, LTE_PPUS_CODE_DEFS SCD WHERE C.CODE = CV.CODE
  AND CV.CODE = SCD.CODE AND C.NO_DELETE_FLAG = 1 AND LOWER (C.NAME) LIKE
  '%other%' UNION SELECT CVI.CODE, CI.CODESET,
  CI.PARENT_NAME||'#'||CI.MODELNUMBER AS NAME FROM XYZ_CODE_ITEMNUMBERS
  CI, XYZ_CODEVIEW_ITEMNUMBERS CVI, LTE_PPUS_CODE_DEFS SCD WHERE
  CI.CODE = CVI.CODE AND CVI.CODE = SCD.CODE AND CI.NO_DELETE_FLAG = 1 AND
  LOWER (CI.MODELNUMBER) LIKE '%other%' AND LOWER(CI.PARENT_NAME) LIKE
  '%other%') OCC WHERE FD.DIM_LEVEL IN (SELECT DIMENSIONVIEW_LEVEL_ID FROM
  XYZ_DIMENSIONVIEW_LEVELS WHERE DIMENSIONVIEW = :B1 ) AND
  FD.INPUT_DATA_ELEMENT_ID = DD.BUSDS_ELEMENT_ID AND DD.CODESET_MAP_ID =
  CM.CODESET_MAP_ID AND CM.CODESET = CC.CODESET AND CC.DERIVEDCODESET = 1 AND
  CC.CODESET = DC.DERIVEDCODESET AND DC.CODESET = OCC.CODESET ) WHERE CODE =
  :B2
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	350	0.16	0.16	0	0	0	0
Fetch	350	142.80	142.83	0	1250690	0	0
total	701	142.97	143.00	0	1250690	0	0

Conclusion

- Take advantage of the available diagnostic and tuning tools
- Don’t forget the fundamental tool kit:
  - Top
  - Explain plan
  - 10046 trace and tkprof
  - Optimizer hint
- Prerequisite: Statistics is up to date

SQL PROFILE and HINT

- To disable the sql profile:

exec dbms_sqltune.alter_sql_profile('SQLPROFILE_01','STATUS','DISABLED')

- To drop the sql profile:

exec dbms_sqltune.drop_sql_profile('SQLPROFILE_01')