ORACLE

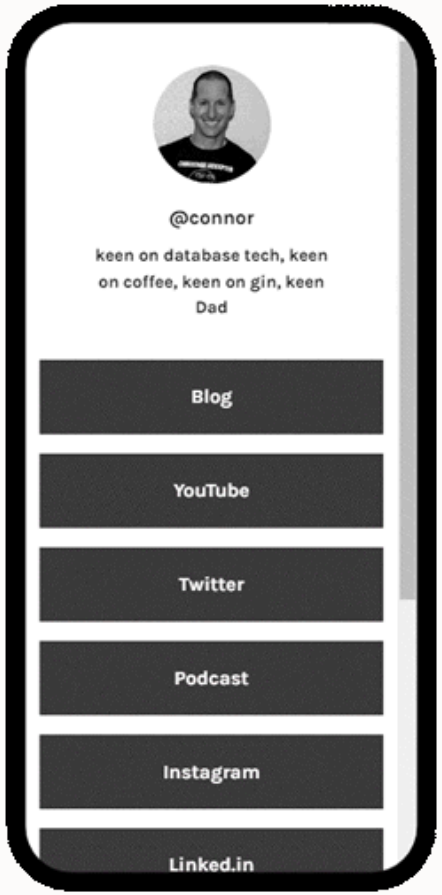
A fresh look at SQL tuning

Connor McDonald
Database Advocate

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Getting in touch is easy...



@connor_mc_d

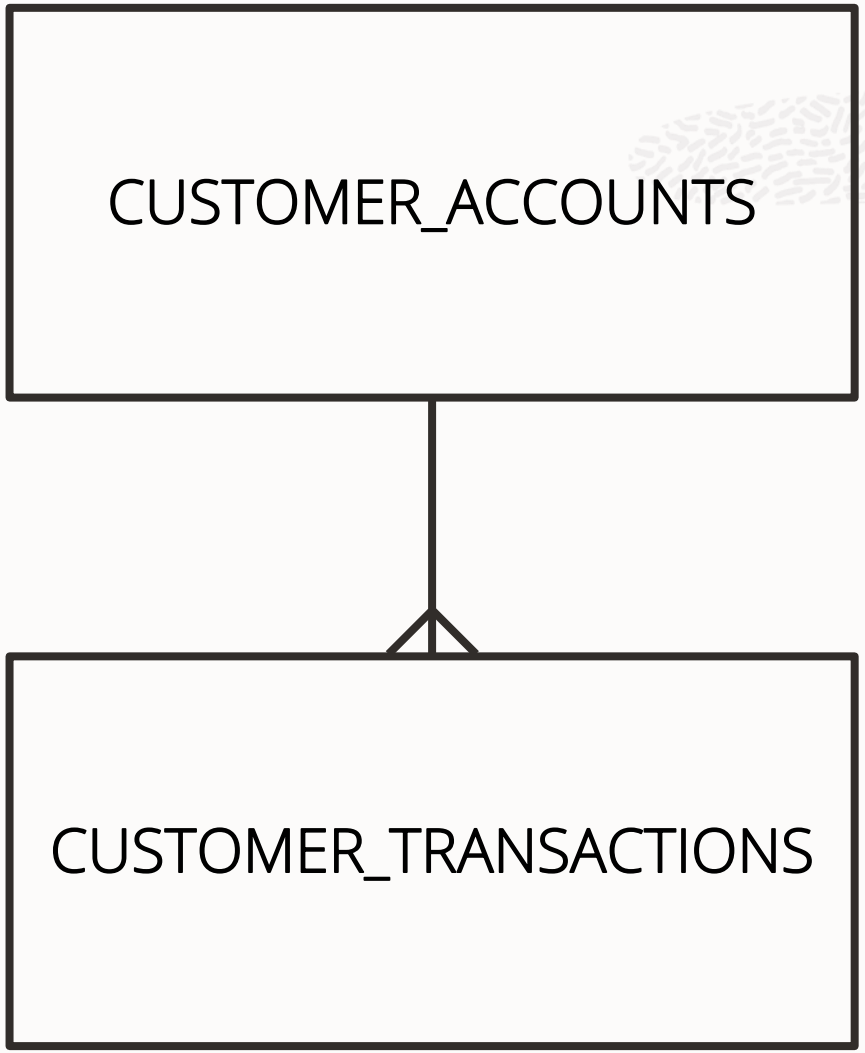
https://linktr.ee/connor

if you are tuning SQL …

#FAIL

why ?

real example


CUSTOMER_ACCOUNTS
CUSTOMER_TRANSACTIONS

"last transaction for all customers"

Account Usage Summary

Bank Account#	Last Transaction	Balance
3005519	2012-10-16 22:50:49	26.00
3005527	2012-10-16 21:00:44	554.77
3005550	2012-10-16 23:17:37	327.90
3005592	2012-10-16 17:20:46	1364.48
3005600	2012-10-16 21:53:40	287.83
3005618	2012-10-16 14:39:12	493.83
3005642	2012-10-16 13:05:29	1895.36
3005667	2012-10-16 11:40:45	3470.92
3005675	2012-10-16 11:36:46	1021.29
3005683	2012-10-16 23:30:17	2918.26

[snip]

Printed: 08:15am, October 17

Requested by: cmcdonald

```
SQL> select ACCOUNT_NUM,
  2         max(TXN_TS) LAST_TS
  3  from   CUSTOMER_TRANS t
  4  group by ACCOUNT_NUM
  5  order by 1;
```

make it faster …

we tried …

```
SQL> select /*+ PARALLEL(t 16) */
  2         ACCOUNT_NUM, max(TXN_TS) LAST_TS
  3  from   CUSTOMER_TRANS t
  4  group by ACCOUNT_NUM
  5  order by 1;
```

worked …

… storage meltdown

```
SQL> create index CUSTOMER_TRANS_IX
  2     on CUSTOMER_TRANS( account_num, txn_ts )

SQL> select ACCOUNT_NUM, max(TXN_TS) LAST_TS
 2   from   CUSTOMER_TRANS t
 3   group by ACCOUNT_NUM
 4   order by 1;

-------------------------------------------------------
| Id  | Operation             | Name                  |
-------------------------------------------------------
|   0 | SELECT STATEMENT      |                       |
|   1 |  INDEX FAST FULL SCAN| CUSTOMER_TRANS_IX |
-------------------------------------------------------
```

worked …

… oltp impact

```
SQL> create materialized view CUSTOMER_SUMMARY
  2  refresh complete on demand as
  3  select ACCOUNT_NUM, max(TXN_TS) LAST_TS
  4  from   CUSTOMER_TRANS t
  5  group by ACCOUNT_NUM
  6  order by 1;
```

worked …

"it doesn't look right"

```
CREATE TABLE CUSTOMER_TRANS
(
   ...
)
PARTITION BY RANGE (TXN_TS)
INTERVAL( NUMTOYMINTERVAL(1,'MONTH'))
(
PARTITION P1 VALUES LESS THAN (TIMESTAMP' 2018-01-01 00:00:00')
PARTITION P2 VALUES LESS THAN (TIMESTAMP' 2018-06-01 00:00:00')
   ...
);
```

```
SQL> select ACCOUNT_NUM, max(TXN_TS) LAST_TS
  2   from   CUSTOMER_TRANS t
  3   where  TXN_TS > add_months(sysdate,-12)
  4   group by ACCOUNT_NUM
  5   order by 1;
```

```
-------------------------------------------------------------------
| Id  | Operation                | Name           |Pstart| Pstop |
-------------------------------------------------------------------
|   0 | SELECT STATEMENT         |                |      |       |
|   1 |  PARTITION RANGE ITER..  |                |  127 |   138 |
|*  2 |   TABLE ACCESS FULL      | CUSTOMER_TRANS |  127 |   138 |
-------------------------------------------------------------------
```

worked …

… changed the function

Report: REP-020C

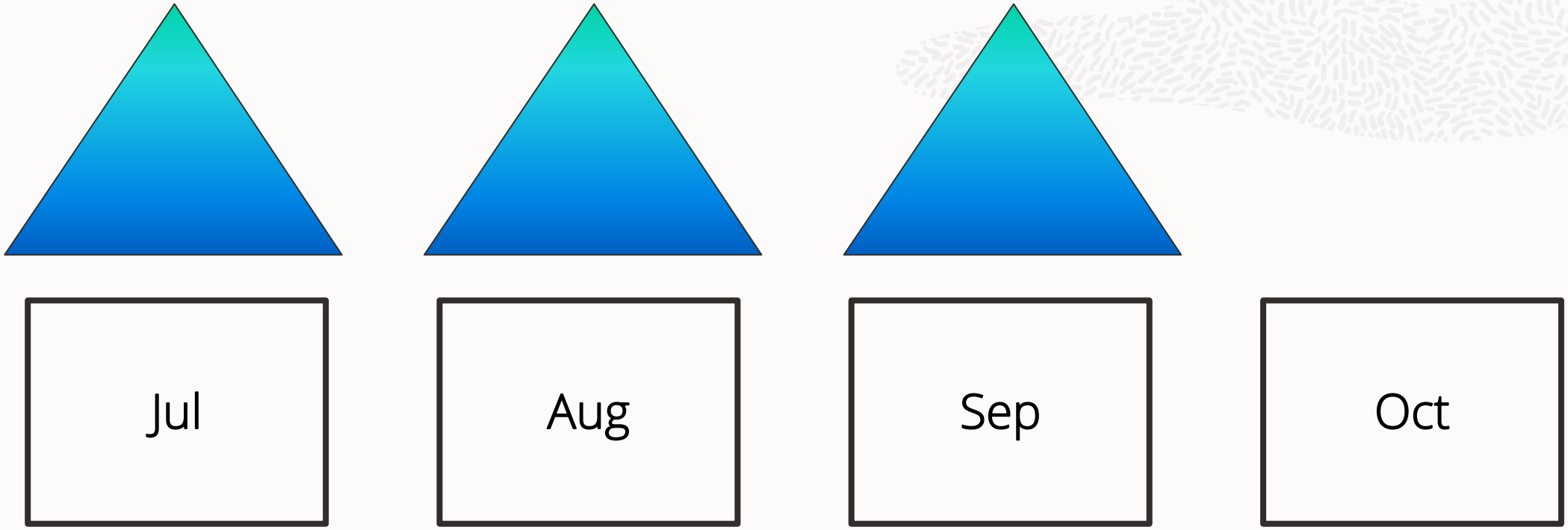
Account Usage Summary

Bank Account#	Last Transaction	Balance
3005519	2012-10-16 22:50:49	26.00
3005527	2012-10-16 21:00:44	554.77
3005550	2012-10-16 23:17:37	327.90
3005592	[illegible]	1364.48
3005600	*>12 mths ago*	287.83
3005618	[illegible]	493.83
3005642	2012-10-16 13:05:29	1895.36
3005667	[illegible]	3470.92
3005675	*>12 mths ago*	1021.29
3005683	[illegible]	2918.26

[snip]

Printed: 08:15am, October 17, 2012

Requested by: cmcdonald


Jul
Aug
Sep
Oct

```
SQL> insert into GTT_LAST_TXN
 2   select ACCOUNT_NUM, max(TXN_TS) LAST_TS
 3   from   CUSTOMER_TRANSACTIONS t
 4   where  TXN_TS >= trunc(sysdate,'MM')
 5   group by ACCOUNT_NUM;

for i in (
  select ACCOUNT_NUM from ACCOUNTS
  minus
  select ACCOUNT_NUM from GTT_LAST_TXN
  )

    insert into GTT_LAST_TXN
    select /*+ INDEX_DESC(t (account_num, txn_ts)) */ MAX(TXN_TS)
    from   CUSTOMER_TRANSACTIONS t
    where  ACCOUNT_NUM = i.ACCOUNT_NUM
    and    TXN_TS < trunc(sysdate,'MM')
```

worked …

… complexity !

problem


```
SQL> select ...
  2    from   ...
  3    where  ...
```

the requirement

"last transaction for all customers"

"we have an interest in recent activity for customers"

promotional material

"we haven't seen you in a while, here's a discount on ...."

the solution ?

```
SQL> desc CUSTOMER_ACCOUNTS

 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 ACCOUNT_NUM                               NOT NULL NUMBER(8)
 CURRENT_BALANCE                           NOT NULL NUMBER(14,2)
 ...
 ...
 ...
 MODIFICATION_TS                           NOT NULL TIMESTAMP(6)
```

optimistic locking

```
select ..., MODIFICATION_TS
into   ..., :prev_modtstamp
from   CUSTOMER_ACCOUNTS
where  ...
```



```
   update CUSTOMER_ACCOUNTS
   set    ...
   where  ACCOUNT_NUM = :b1
   and    MODIFICATION_TS = :prev_modtstamp

   if SQL%NOTFOUND then
      msg('Row has been changed by another user, please requery');
   end if;
```

```
SQL> select ACCOUNT_NUM,
  2         MODIFICATION_TS approx_last_usage
  3  from   CUSTOMER_ACCOUNTS
```

that's it !

don't focus on the SQL

don't tune SQL ...

… tune this instead

HERE

UX

sql tuning is ...

warning ☺


mm
0.5mm
1
2
3
4
5
6
7
8
9
10
11
12

stop the bleeding...

...until find a cure


Emergency Ambulance

```
SQL> select sql_fulltext
  2  from   v$sql
  3  where  module = 'PAYROLL'
  4  and    action = 'Process Bonus'
  5  and    ( buffer_gets > 1000000 or
  6             executions  > 10000 or
  7             disk_reads  > 100000
  8  );

SQL_FULLTEXT
--------------------------------------------------
SELECT ...
FROM   ...
```

```
SQL> select sql_fulltext
  2  from   v$sql
  3  where  module = 'PAYROLL'
  4  and    action = 'Process Bonus'
  5  and    ( buffer_gets > 1000000 or
  6             executions  > 10000 or
  7             disk_reads  > 100000
  8  );

SQL_FULLTEXT
--------------------------------------------------
SELECT ...
FROM   ...
```


```
SQL> select sql_fulltext
  2  from   v$sql
  3  where  module = 'PAYROLL'
  4  and    action = 'Process Bonus'
  5  and    ( buffer_gets > 1000000 or
  6             executions  > 10000 or
  7             disk_reads  > 100000
  8  );

SQL_FULLTEXT
--------------------------------------------------
SELECT ...
FROM   ...
```

latching

V$SQL is the database's SQL

hunting for bad SQL …

… with a bad SQL

you can do better

```
SQL> select sql_fulltext
  2  from   v$sqlstats
  3  where  module = 'PAYROLL'
  4  and    action = 'Process Bonus'
  5  and    ( buffer_gets > 1000000 or
  6             executions  > 10000 or
  7             disk_reads  > 100000
  8  );

SQL_FULLTEXT
--------------------------------------------------
SELECT ...
FROM   ...
```

"The column definitions for columns in V$SQLSTATS are identical to those in the V$SQL and V$SQLAREA views. However, the V$SQLSTATS view differs from V$SQL and V$SQLAREA in that it is faster, more scalable, and has a greater data retention (the statistics may still appear in this view, even after the cursor has been aged out of the shared pool)."

its not the (real) shared pool

if lacking instrumentation…

DBMS_MONITOR

SQL trace is your friend

DBMS_USERDIAG

found your sql...

```
SQL> select count(*)
  2  from   VEHICLES
  3  where  MAKE  = 'HONDA'
  4  and    MODEL = 'CIVIC';
```

1) what is the real query

```
SQL> set autotrace traceonly stat

SQL> select *
  2  from   LOOKS_SO_INNOCENT
  3  where  CREATED > sysdate

Statistics
----------------------------------------------------------
        651  recursive calls
          0  db block gets
      23243  consistent gets
         24  physical reads
```

```
SQL> variable c clob

SQL> begin
  2    dbms_utility.expand_sql_text
  3     ( 'select * from LOOKS_SO_INNOCENT '||
  4       'where  created > sysdate',:c);
  5  end;
  6  /

PL/SQL procedure successfully completed.

SQL> print c
```

```
SQL> create or replace
  2  view LOOK_SO_INNOCENT as
  3  select
  4    o.owner,
  5    o.created,
  6    s.bytes,
  7    s.tablespace_name
  8  from
  9    dba_segments s,
 10    all_objects o
 11  where o.owner = s.owner
 12  and   o.object_name = s.segment_name;

View created.
```

```
SELECT "A1"."OWNER" "OWNER",
"A1"."NAME" "NAME",
"A1"."CREATED" "CREATED",
"A1"."BYTES" "BYTES",
"A1"."TABLESPACE_NAME" "TABLESPACE_NAME"
FROM  (SELECT "A2"."OWNER" "OWNER",
"A2"."OBJECT_NAME" "NAME",
"A2"."CREATED" "CREATED",
"A3"."BYTES" "BYTES",
"A3"."TABLESPACE_NAME" "TABLESPACE_NAME" FROM  (SELECT "A4"
."OWNER" "OWNER",
"A4"."SEGMENT_NAME" "SEGMENT_NAME",
"A4"."PARTITION_NAME" "PARTITION_NAME",
"A4"."SEGMENT_TYPE" "SEGMENT_TYPE",
"A4"."SEGMENT_SUBTYPE" "SEGMENT_SUBTYPE",
"A4"."TABLESPACE_NAME" "TABLESPACE_NAME",
"A4"."HEADER_FILE" "HEADER_FILE",
"A4"."HEADER_BLOCK" "HEADER_BLOCK",
DECODE(BITAND("A4"."SEGMENT_FLAGS",131072),131072,
"A4"."BLOCKS",DECODE(BITAND("A4"."SEGMENT_FLAGS",1),1,
"SYS"."DBMS_SPACE_ADMIN"."SEGMENT_NUMBER_BLOCKS"("A4"."TABLESPACE_ID",
"A4"."RELATIVE_FNO",
"A4"."HEADER_BLOCK",
"A4"."SEGMENT_TYPE_ID",
"A4"."BUFFER_POOL_ID",
"A4"."SEGMENT_FLAGS",
"A4"."SEGMENT_OBJD",
"A4"."BLOCKS"),
"A4"."BLOCKS"))*"A4"."BLOCKSIZE" "BYTES",
DECODE(BITAND("A4"."SEGMENT_FLAGS",131072),131072,
"A4"."BLOCKS",DECODE(BITAND("A4"."SEGMENT_FLAGS",1),1,
"SYS"."DBMS_SPACE_ADMIN"."SEGMENT_NUMBER_BLOCKS"("A4"."TABLESPACE_ID",
"A4"."RELATIVE_FNO",
```

```
"A4"."HEADER_BLOCK",
"A4"."SEGMENT_TYPE_ID",
"A4"."BUFFER_POOL_ID",
"A4"."SEGMENT_FLAGS",
"A4"."SEGMENT_OBJD",
"A4"."BLOCKS"),
"A4"."BLOCKS")) "BLOCKS",DECODE(BITAND("A4"."SEGMENT_FLAGS",131072),131072,
"A4"."EXTENTS",DECODE(BITAND("A4"."SEGMENT_FLAGS",1),1,
"SYS"."DBMS_SPACE_ADMIN"."SEGMENT_NUMBER_EXTENTS"("A4"."TABLESPACE_ID",
"A4"."RELATIVE_FNO",
"A4"."HEADER_BLOCK",
"A4"."SEGMENT_TYPE_ID",
"A4"."BUFFER_POOL_ID",
"A4"."SEGMENT_FLAGS",
"A4"."SEGMENT_OBJD",
"A4"."EXTENTS"),
"A4"."EXTENTS")) "EXTENTS",
"A4"."INITIAL_EXTENT" "INITIAL_EXTENT",
"A4"."NEXT_EXTENT" "NEXT_EXTENT",
"A4"."MIN_EXTENTS" "MIN_EXTENTS",
"A4"."MAX_EXTENTS" "MAX_EXTENTS",
"A4"."MAX_SIZE" "MAX_SIZE",
"A4"."RETENTION" "RETENTION",
"A4"."MINRETENTION" "MINRETENTION",
"A4"."PCT_INCREASE" "PCT_INCREASE",
"A4"."FREELISTS" "FREELISTS",
"A4"."FREELIST_GROUPS" "FREELIST_GROUPS",
"A4"."RELATIVE_FNO" "RELATIVE_FNO",
DECODE("A4"."BUFFER_POOL_ID",1,'KEEP',2,'RECYCLE','DEFAULT') "BUFFER_POOL",
DECODE("A4"."FLASH_CACHE",1,'KEEP',2,
'NONE','DEFAULT') "FLASH_CACHE",DECODE("A4"."CELL_FLASH_CACHE",1,'KEEP',2,'NONE','DEFAULT')
"CELL_FLASH_CACHE"
FROM  ( (SELECT NVL("A199"."NAME",'SYS') "OWNER",
"A198"."NAME" "SEGMENT_NAME",
```

```
"A198"."SUBNAME" "PARTITION_NAME",
"A196"."OBJECT_TYPE" "SEGMENT_TYPE",
"A195"."TYPE#" "SEGMENT_TYPE_ID",DECODE(BIT
AND("A195"."SPARE1",2097408),2097152,'SECUREFILE',256,'ASSM','MSSM') "SEGMENT_SUBTYPE",
"A197"."TS#" "TABLESPACE_ID",
"A197"."NAME" "TABLESPACE_NAME",
"A197"."BLOCKSIZE" "BLOCKSIZE",
"A194"."FILE#" "HEADER_FILE",
"A195"."BLOCK#" "HEADER_BLOCK",
"A195"."BLOCKS"*"A197"."BLOCKSIZE" "BYTES",
"A195"."BLOCKS" "BLOCKS",
"A195"."EXTENTS" "EXTENTS",
"A195"."INIEXTS"*"A197"."BLOCKSIZE" "INITIAL_EXTENT",
"A195"."EXTSIZE"*"A197"."BLOCKSIZE" "NEXT_EXTENT",
"A195"."MINEXTS" "MIN_EXTENTS",
"A195"."MAXEXTS" "MAX_EXTENTS",DECODE(BITAND("A195"."SPARE1",4194304),4194304,
"A195"."BITMAPRANGES",NULL) "MAX_SIZE",TO_CHAR(DECODE(
BITAND("A195"."SPARE1",2097152),2097152,
DECODE("A195"."LISTS",0,'NONE',1,'AUTO',2,'MIN',3,'MAX',4,'DEFAULT','INVALID'),NULL))
"RETENTION",DECODE(BITAND("A195"."SPARE1",2097152),2097152,
"A195"."GROUPS",NULL) "MINRETENTION",DECODE(BITAND("A197"."FLAGS",3),1,TO_NUMBER(NULL),
"A195"."EXTPCT") "PCT_INCREASE",DECODE(BITAND("A197"."FLAGS",32),32,
TO_NUMBER(NULL),DECODE("A195"."LISTS",0,1,
"A195"."LISTS"))
"FREELISTS",DECODE(BITAND("A197"."FLAGS",32),32,TO_NUMBER(NULL),DECODE("A195"."GROUPS",0,1,
"A195"."GROUPS")) "FREELIST_GROUPS",
"A195"."FILE#" "RELATIVE_FNO",BITAND("A195"."CACHEHINT",3) "BUFFER_POOL_ID",
BITAND("A195"."CACHEHINT",12)/4 "FLASH_CACHE",BITAND("A195"."CACHEHINT",48)/16 "CELL_FLASH_CACHE",
NVL("A195"."SPARE1",0)
"SEGMENT_FLAGS",DECODE(BITAND("A195"."SPARE1",1),1,
"A195"."HWMINCR",
"A198"."DATAOBJ#") "SEGMENT_OBJD" FROM "SYS"."USER$" "A199",
"SYS"."OBJ$" "A198",
"SYS"."TS$" "A197", ( (SELECT
```

```
DECODE(BITAND("A209"."PROPERTY",8192),8192,'NESTED TABLE','TABLE') "OBJECT_TYPE",
 2 "OBJECT_TYPE_ID",5 "SEGMENT_TYPE_ID",
"A209"."OBJ#" "OBJECT_ID",
"A209"."FILE#" "HEADER_FILE",
"A209"."BLOCK#" "HEADER_BLOCK",
"A209"."TS#" "TS_NUMBER" FROM "SYS"."TAB$" "A209" WHERE BITAND("A209"."PROPERTY",1024)=0) UNI
ON ALL  (SELECT 'TABLE PARTITION' "OBJECT_TYPE",19 "OBJECT_TYPE_ID",5 "SEGMENT_TYPE_ID",
"A208"."OBJ#" "OBJECT_ID",
"A208"."FILE#" "HEADER_FILE",
"A208"."BLOCK#" "HEADER_BLOCK",
"A208"."TS#" "TS_NUMBER" FROM "SYS"."TABPART$" "A208")  UNION ALL
  (SELECT 'CLUSTER' "OBJECT_TYPE",3 "OBJECT_TYPE_ID",5 "SEGMENT_TYPE_ID",
"A207"."OBJ#" "OBJECT_ID",
"A207"."FILE#" "HEADER_FILE",
"A207"."BLOCK#" "HEADER_BLOCK",
"A207"."TS#" "TS_NUMBER" FROM "SYS"."CLU$" "A207")  UNION ALL
  (SELECT DECODE("A206"."TYPE#",8,'LOBINDEX','INDEX') "OBJECT_TYPE",1 "OBJECT_TYPE_ID",6
"SEGMENT_TYPE_ID",
"A206"."OBJ#" "OBJECT_ID",
"A206"."FILE#" "HEADER_FILE",
"A206"."BLOCK#" "HEADER_BLOCK",
"A206"."TS#" "TS_NUMBER" FROM "SYS"."IND$" "A206" WHERE "A206"."TYPE#"=1 OR
"A206"."TYPE#"=2 OR
"A206"."TYPE#"=3 OR
"A206"."TYPE#"=4 OR
"A206"."TYPE#"=6 OR
"A206"."TYPE#"=7 OR
"A206"."TYPE#"=8 OR
"A206"."TYPE#"=9)  UNION ALL
  (SELECT 'INDEX PARTITION'
 "OBJECT_TYPE",20 "OBJECT_TYPE_ID",6 "SEGMENT_TYPE_ID",
"A205"."OBJ#" "OBJECT_ID",
"A205"."FILE#" "HEADER_FILE",
"A205"."BLOCK#" "HEADER_BLOCK",
```

```
"A205"."TS#" "TS_NUMBER" FROM "SYS"."INDPART$" "A205")  UNION ALL
  (SELECT 'LOBSEGMENT' "OBJECT_TYPE",21 "OBJECT_TYPE_ID",8 "SEGMENT_TYPE_ID",
"A204"."LOBJ#" "OBJECT_ID",
"A204"."FILE#" "HEADER_FILE",
"A204"."BLOCK#" "HEADER_BLOCK",
"A204"."TS#" "TS_NUMBER" FROM "SYS"."LOB$" "A204" WHERE BITAND("A204"."PROPERTY",64)=0 OR
BITAND("A204"."PROPERTY",128)=128)  UNION ALL
  (SELECT 'TABLE SUBPARTITION' "OBJECT_TYPE",34 "OBJECT_TYPE_ID",5 "SEGMENT_TYPE_ID",
"A203"."OBJ#" "OBJECT_ID",
"A203"."FILE#" "HEADER_FILE",
"A203"."BLOCK#" "HEADER_BLOCK",
"A203"."TS#" "TS_NUMBER" FROM "SYS"."TABSUBPART$" "A203")  UNION ALL
  (SELECT 'INDEX SUBPARTITION' "OBJECT_TYPE",35 "OBJECT_TYPE_ID",6 "SEGMENT_TYPE_ID",
"A202"."OBJ#" "OBJECT_ID",
"A202"."FILE#" "HEADER_FILE",
"A202"."BLOCK#" "HEADER_BLOCK",
"A202"."TS#" "TS_NUMBER" FROM "SYS"."INDSUBPART$" "A202")  UNION ALL
  (SELECT DECODE("A201"."FRAGTYPE$",'P','LOB PARTITION','LOB SUBPARTITION')
  "OBJECT_TYPE",DECODE("A201"."FRAGTYPE$",'P',40,41) "OBJECT_TYPE_ID",8 "SEGMENT_TYPE_ID",
"A201"."FRAGOBJ#" "OBJECT_ID",
"A201"."FILE#" "HEADER_FILE",
"A201"."BLOCK#" "HEADER_BLOCK",
"A201"."TS#" "TS_NUMBER" FROM "SYS"."LOBFRAG$" "A201")) "A196",
"SYS"."SEG$" "A195",
"SYS"."FILE$" "A194" WHERE "A195"."FILE#"="A196"."HEADER_FILE"  AND
"A195"."BLOCK#"="A196"."HEADER_BLOCK"  AND
"A195"."TS#"="A196"."TS_NUMBER"  AND
"A195"."TS#"="A197"."TS#"  AND
"A198"."OBJ#"="A196"."OBJECT_ID"  AND
"A198"."OWNER#"="A199"."USER#"(+)  AND
"A195"."TYPE#"="A196"."SEGMENT_TYPE_ID"  AND
"A198"."TYPE#"="A196"."OBJECT_TYPE_ID"  AND
"A195"."TS#"="A194"."TS#"  AND
```

```
"A195"."FILE#"="A194"."RELFILE#")  UNION ALL
  (SELECT NVL("A193"."NAME",'SYS') "OWNER",
"A191"."NAME" "SEGMENT_NAME",NULL "PARTITION_NAME",
DECODE("A190"."TYPE#",1,'ROLLBACK',10,'TYPE2 UNDO') "SEGMENT_TYPE",
"A190"."TYPE#" "SEGMENT_TYPE_ID",NULL "SEGMENT_SUBTYPE",
"A192"."TS#" "TABLESPACE_ID",
"A192"."NAME" "TABLESPACE_NAME",
"A192"."BLOCKSIZE" "BLOCKSIZE",
"A189"."FILE#" "HEADER_FILE",
"A190"."BLOCK#" "HEADER_BLOCK",
"A190"."BLOCKS"*"A192"."BLOCKSIZE" "BYTES",
"A190"."BLOCKS" "BLOCKS",
"A190"."EXTENTS" "EXTENTS",
"A190"."INIEXTS"*"A192"."BLOCKSIZE" "INITIAL_EXTENT",
"A190"."EXTSIZE"*"A192"."BLOCKSIZE" "NEXT_EXTENT",
"A190"."MINEXTS" "MIN_EXTENTS",
"A190"."MAXEXTS" "MAX_EXTENTS",DECODE(BITAND("A190"."SPARE1",4194304),4194304,
"A190"."BITMAPRANGES",NULL) "MAX_SIZE",NULL "RETENTION",NULL "MINRETENTION",
"A190"."EXTPCT"
"PCT_INCREASE",DECODE(BITAND("A192"."FLAGS",32),32,TO_NUMBER(NULL),DECODE("A190"."LISTS",0,1,
"A190"."LISTS")) "FREELISTS",DECODE(BITAND("A192"."FLAGS",32),32,TO_NUMBER(NULL),
DECODE("A190"."GROUPS",0,1,"A190"."GROUPS")) "FREELIST_GROUPS",
"A190"."FILE#" "RELATIVE_FNO",BITAND("A190"."CACHEHINT",3)
"BUFFER_POOL_ID",BITAND("A190"."CACHEHINT",12)/4 "FLASH_CACHE",
BITAND("A190"."CACHEHINT",48)/16 "CELL_FLASH_CACHE",NVL("A190"."SPARE1",0) "SEGMENT_FLAGS",
"A191"."US#" "SEGMENT_OBJD" FROM "SYS"."USER$" "A193","SYS"."TS$" "A192",
"SYS"."UNDO$" "A191",
"SYS"."SEG$" "A190",
"SYS"."FILE$" "A189" WHERE "A190"."FILE#"="A191"."FILE#"  AND
"A190"."BLOCK#"="A191"."BLOCK#"  AND
"A190"."TS#"="A191"."TS#"  AND
"A190"."TS#"="A192"."TS#"  AND
"A190"."USER#"="A193"."USER#"(+)  AND
("A190"."TYPE#"=1 OR
"A190"."TYPE#"=10)  AND
"A191"."STATUS$"<>1  AND
```

```
"A191"."TS#"="A189"."TS#"  AND
"A191"."FILE#"="A189"."RELFILE#")  UNION ALL
  (SELECT NVL("A188"."NAME",'SYS') "OWNER",
  TO_CHAR("A185"."FILE#")||'.'||TO_CHAR("A186"."BLOCK#") "SEGMENT_NAME",NULL "PARTITION_NAME",
  DECODE("A186"."TYPE#",2,'DEFERRED ROLLBACK',3,
  'TEMPORARY',4,'CACHE',9,'SPACE HEADER','UNDEFINED') "SEGMENT_TYPE",
"A186"."TYPE#" "SEGMENT_TYPE_ID",NULL "SEGMENT_SUBTYPE",
"A187"."TS#" "TABLESPACE_ID",
"A187"."NAME" "TABLESPACE_NAME",
"A187"."BLOCKSIZE" "BLOCKSIZE",
"A185"."FILE#" "HEADER_FILE",
"A186"."BLOCK#" "HEADER_BLOCK",
"A186"."BLOCKS"*"A187"."BLOCKSIZE" "BYTES",
"A186"."BLOCKS" "BLOCKS",
"A186"."EXTENTS" "EXTENTS",
"A186"."INIEXTS"*"A187"."BLOCKSIZE" "INITIAL_EXTENT",
"A186"."EXTSIZE"*"A187"."BLOCKSIZE" "NEXT_EXTENT",
"A186"."MINEXTS" "MIN_EXTENTS",
"A186"."MAXEXTS" "MAX_EXTENTS",DECODE(BITAND("A186"."SPARE1",4194304),4194304,
"A186"."BITMAPRANGES",NULL) "MAX_SIZE",NULL
"RETENTION",NULL "MINRETENTION",DECODE(BITAND("A187"."FLAGS",3),1,TO_NUMBER(NULL),
"A186"."EXTPCT")
"PCT_INCREASE",DECODE(BITAND("A187"."FLAGS",32),32,TO_NUMBER(NULL),DECODE("A186"."LISTS",0,1,
"A186"."LISTS"))
"FREELISTS",DECODE(BITAND("A187"."FLAGS",32),32,TO_NUMBER(NULL),DECODE("A186"."GROUPS",0,1,
"A186"."GROUPS")) "FREELIST_GROUPS",
"A186"."FILE#" "RELATIVE_FNO",BITAND("A186"."CACHEHINT",3)
"BUFFER_POOL_ID",BITAND("A186"."CACHEHINT",12)/4 "FLASH_CACHE",BITAND("A186"."CACHEHINT",48)/16
"CELL_FLASH_CACHE",NVL("A186"."SPARE1",0) "SEGMENT_FLAGS",
"A186"."HWMINCR" "SEGMENT_OBJD" FROM "SYS"."USER$"
"A188",
"SYS"."TS$" "A187",
"SYS"."SEG$" "A186",
"SYS"."FILE$" "A185" WHERE "A186"."TS#"="A187"."TS#"  AND
"A186"."USER#"="A188"."USER#"(+)  AND
"A186"."TYPE#"<>1  AND
"A186"."TYPE#"<>5  AND
```

```
"A186"."TYPE#"<>6  AND
"A186"."TYPE#"<>8  AND
"A186"."TYPE#"<>10  AND
"A186"."TYPE#"<>11  AND
"A186"."TS#"="A185"."TS#"
 AND
"A186"."FILE#"="A185"."RELFILE#")  UNION ALL
  (SELECT NVL("A184"."NAME",'SYS') "OWNER",'HEATMAP' "SEGMENT_NAME",
  NULL "PARTITION_NAME",'SYSTEM STATISTICS' "SEGMENT_TYPE",
"A182"."TYPE#" "SEGMENT_TYPE_ID",NULL "SEGMENT_SUBTYPE",
"A183"."TS#" "TABLESPACE_ID",
"A183"."NAME" "TABLESPACE_NAME",
"A183"."BLO
CKSIZE" "BLOCKSIZE",
"A181"."FILE#" "HEADER_FILE",
"A182"."BLOCK#" "HEADER_BLOCK",
"A182"."BLOCKS"*"A183"."BLOCKSIZE" "BYTES",
"A182"."BLOCKS" "BLOCKS",
"A182"."EXTENTS" "EXTENTS",
"A182"."INIEXTS"*"A183"."BLOCKSIZE" "INITIAL_EXTENT",
"A182"."EXTSIZE"*"A183"."BLOCKSIZE" "NEXT_EXTENT",
"A182"."MINEXTS" "MIN_EXTENTS",
"A182"."MAXEXTS" "MAX_EXTENTS",DECODE(BITAND("A182"."SPARE1",4194304),4194304,
"A182"."BITMAPRANGES",NULL) "MAX_SIZE",NULL "RETENTION",NULL
"MINRETENTION",DECODE(BITAND("A183"."FLAGS",3),1,TO_NUMBER(NULL),
"A182"."EXTPCT") "PCT_INCREASE",DECODE(BITAND("A183"."FLAGS",32),32,TO_NUMBER(NULL),DEC
ODE("A182"."LISTS",0,1,
"A182"."LISTS"))
"FREELISTS",DECODE(BITAND("A183"."FLAGS",32),32,TO_NUMBER(NULL),DECODE("A182"."GROUPS",0,1,
"A182"."GROUPS")) "FREELIST_GROUPS",
"A182"."FILE#" "RELATIVE_FNO",BITAND("A182"."CACHEHINT",3) "BUFFER_POOL_ID",
BITAND("A182"."CACHEHINT",12)/4 "FLASH_CACHE",BITAND("A18
2"."CACHEHINT",48)/16 "CELL_FLASH_CACHE",NVL("A182"."SPARE1",0) "SEGMENT_FLAGS",
"A182"."HWMINCR" "SEGMENT_OBJD" FROM "SYS"."USER$" "A184",
```

```
"SYS"."TS$" "A183",
"SYS"."SEG$" "A182",
"SYS"."FILE$" "A181" WHERE "A182"."TS#"="A183"."TS#"  AND
"A182"."USER#"="A184"."USER#"(+)  AND
"A182"."TYPE#"=11  AND
"A182"."TS#"="A181"."TS#"  AND
"A182"."FILE#"="A181"."RELFILE#")) "A4") "A3", (SELECT "A5"."NAME" "OWNER",
"A6"."NAME" "OBJECT_NAME",
"A6"."SUBNAME" "SUBOBJECT_NAME",
"A6"."OBJ#" "OBJECT_ID",
"A6"."DATAOBJ#" "DATA_OBJECT_ID",DECODE("A6"."TYPE#",0,'NEXT OBJECT',1,'INDEX',
2,'TABLE',3,'CLUSTER',4,'VIEW',5,'SYNONYM
',6,'SEQUENCE',7,'PROCEDURE',8,'FUNCTION',9,'PACKAGE',11,'PACKAGE BODY',12,
'TRIGGER',13,'TYPE',14,'TYPE BODY',19,'TABLE PARTITION',20,'INDEX PARTITION',
21,'LOB',22,'LIBRARY',23,'DIRECTORY',24,'QUEUE',28,'JAVA SOURCE',29,'JAVA CLASS',30,
'JAVA RESOURCE',32,'INDEXTYPE',33,'OPERATOR',34,
'TABLE SUBPARTITION',35,'INDEX SUBPARTITION',40,'LOB PARTITION',41,'LOB SUBPARTITION',42
,NVL( (SELECT 'REWRITE EQUIVALENCE' "'REWRITEEQUIVALENCE'" FROM SYS."SUM$" "A52" WHERE "A52"."OBJ#"="A6"."OBJ#"  AND
BITAND("A52"."XPFLAGS",8388608)=8388608),'MATERIALIZED VIEW'),43,'DIMENSION',
44,'CONTEXT',46,'RULE SET',47,'RESOURCE PLAN',48,'CONSUMER GROUP',55,'XML SCHEMA',56,'JAVA
DATA',57,'EDITION',59,'RULE',
60,'CAPTURE',61,'APPLY',62,'EVALUATION CONTEXT',66,'JOB',67,'PROGRAM',68,'JOB CLASS',69,
'WINDOW',72,'SCHEDULER GROUP',74,'SCHEDULE',79,'CHAIN',81,'FILE GROUP',82,'MINING
MODEL',87,'ASSEMBLY',90,'CREDENTIAL',92,'CUBE
 DIMENSION',93,'CUBE',94,'MEASURE FOLDER',95,'CUBE BUILD PROCESS',100,'FILE WATCHER',
 101,'DESTINATION',114,'SQL TRANSLATION PROFILE',115,'UNIFIED AUDIT POLICY','UNDEFINED') "OBJECT_TYPE",
"A6"."CTIME" "CREATED",
"A6"."MTIME" "LAST_DDL_TIME",TO_CHAR("A6"."STIME",'YYYY-MM-DD:HH24:MI:SS') "TIMESTAMP",DEC
ODE("A6"."STATUS",0,'N/A',1,'VALID','INVALID') "STATUS",DECODE(BITAND("A6"."FLAGS",2),0,'N',2,'Y','N')
"TEMPORARY",DECODE(BITAND("A6"."FLAGS",4),0,'N',4,'Y','N')
"GENERATED",DECODE(BITAND("A6"."FLAGS",16),0,'N',16,'Y','N') "SECONDARY",
"A6"."NAMESPACE" "NAMESPACE",
"A6"."DEFINING_EDITION" "EDITION_NAM
E",DECODE(BITAND("A6"."FLAGS",196608),65536,'METADATA LINK',131072,'OBJECT LINK','NONE') "SHARING",
CASE  WHEN ("A6"."TYPE#"=4 OR
"A6"."TYPE#"=5 OR
```

16 …

more …

pages !

```
SQL> select count(*)
  2  from   VEHICLES
  3  where  MAKE  = 'HONDA'
  4  and    MODEL = 'CIVIC';
```


back to our problem SQL

2) is it sensible/correct ?

```
SQL> select DISTINCT ...
  2  from    EMP e,
  3          DEPT d
  4  where   e.JOB = 'SALES'
  5  and     d.LOC = 'NORTH';
```

```
SQL> select REGION, min(AMOUNT)
  2  from   EMP e,
  3         SALES_TRANSACTIONS s
  4  where  e.JOB      = 'SALES'
  5  and    s.EMPNO    = e.EMPNO
  6  and    s.REGION   = 'CA'
  7  and    s.TAX_AMT > 10
  8  or     s.SUBSIDY > 0
  9  group by REGION
```

```
SQL> select *
  2  from   sales
  3  where  region in
  4    ( select region
  5      from   promoted_locations
  6      where  campaign = 'BLACK FRIDAY');
```

```
SQL> desc promoted_locations
 Name                          Null?    Type
 ----------------------------- -------- -------------
 ID                            NOT NULL NUMBER
 CAMPAIGN                      NOT NULL VARCHAR2(128)
 PROMOTED_REGION               NOT NULL VARCHAR2(24)
 ...
```

aliases aren't just for parsing

"Phone-based" SQL Plan Management

Manual SQL Plan Management

```
SQL> exec DBMS_SPM.LOAD_PLANS_FROM_AWR

SQL> exec DBMS_SPM.CREATE_EVOLVE_TASK

SQL> exec DBMS_SPM.EVOLVE_SQL_PLAN_BASELINE
```

Automatic SQL Plan Management

```
DBMS_SPM.CONFIGURE('AUTO_SPM_EVOLVE_TASK','ON');

DBMS_SPM.SET_EVOLVE_TASK_PARAMETER(
      task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
      parameter => 'ALTERNATE_PLAN_SOURCE',
      value     => 'AUTO');
```

23ai Real Time SQL Plan Management

Q&A
Learn More

Web: oracle.com/enterprisemanager

Videos: youtube.com/OracleEnterpriseMgr

Blogs: blogs.oracle.com/observability

Docs: http://docs.oracle.com/en/enterprise-manager

Try it now

Hands-on-labs

Oracle Cloud Free Tier

Always Free
Services you can use for unlimited time

+

30-Day Free Trial
Free credits you can use for more services

www.oracle.com/cloud/free