



SAGE Computing Services

Customised Oracle Training Workshops and Consulting

New Tuning Features in Oracle 11g - How to make your database as boring as possible

Penny Cookson
Managing Director

Automatic Diagnostic Repository (ADR)

File based repository of diagnostic information including:-

- trace files

- alert log

- dump files

- health monitoring reports

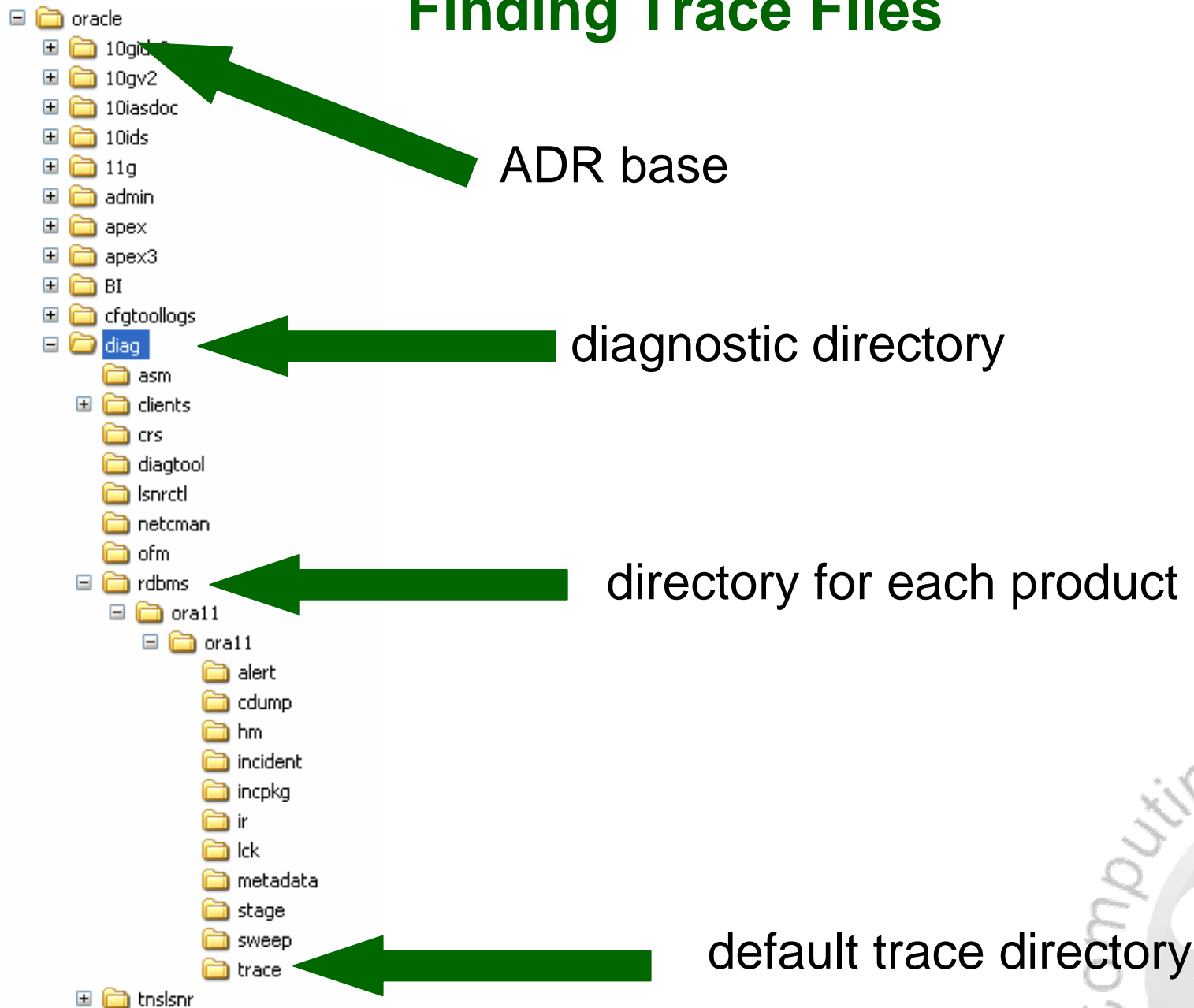
- incidents and problem descriptions

Command line interface ADRCI

Enterprise Manager Support Workbench



Finding Trace Files



Finding Trace Files

```
SELECT      name, description, value , isdeprecatd
FROM        v$parameter
WHERE       name = 'diagnostic_dest'
OR          name like '%dump%'
```

| NAME | DESCRIPTION | VALUE | ISDEPRECATED |
|----------------------|--------------------------------------|--|--------------|
| shadow_core_dump | Core Size for Shadow Processes | none | FALSE |
| background_core_dump | Core Size for Background Processes | partial | FALSE |
| background_dump_dest | Detached process dump directory | c:\oracle\diag\rdbms\ora11\ora11\trace | TRUE |
| user_dump_dest | User process dump directory | c:\oracle\diag\rdbms\ora11\ora11\trace | TRUE |
| core_dump_dest | Core dump directory | c:\oracle\diag\rdbms\ora11\ora11\cdump | FALSE |
| diagnostic_dest | diagnostic base directory | C:\ORACLE | FALSE |
| max_dump_file_size | Maximum size (in bytes) of dump file | unlimited | FALSE |

Finding Trace Files

```
SELECT *  
FROM v$diag_info
```

| INST_ID | NAME | VALUE |
|---------|-----------------------|---|
| 1 | Diag Enabled | TRUE |
| 1 | ADR Base | c:\oracle |
| 1 | ADR Home | c:\oracle\diag\rdbms\ora11\ora11 |
| 1 | Diag Trace | c:\oracle\diag\rdbms\ora11\ora11\trace |
| 1 | Diag Alert | c:\oracle\diag\rdbms\ora11\ora11\alert |
| 1 | Diag Incident | c:\oracle\diag\rdbms\ora11\ora11\incident |
| 1 | Diag Cdump | c:\oracle\diag\rdbms\ora11\ora11\cdump |
| 1 | Health Monitor | c:\oracle\diag\rdbms\ora11\ora11\hm |
| 1 | Default Trace File | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_ora_5088.trc |
| 1 | Active Problem Count | 0 |
| 1 | Active Incident Count | 0 |

Finding Trace Files

```
SELECT pid, program, tracefile  
FROM v$process
```

| PID | PROGRAM | TRACEFILE |
|-----|--------------------|--|
| 1 | PSEUDO | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_ora_0.trc |
| 2 | ORACLE.EXE (PMON) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_pmon_4664.trc |
| 3 | ORACLE.EXE (VKT) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_vktm_4660.trc |
| 4 | ORACLE.EXE (DIAG) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_diag_4756.trc |
| 5 | ORACLE.EXE (DBRM) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_dbrm_1264.trc |
| 6 | ORACLE.EXE (PSP0) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_psp0_4796.trc |
| 7 | ORACLE.EXE (SHAD) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_ora_1084.trc |
| 8 | ORACLE.EXE (DIA0) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_dia0_6008.trc |
| 9 | ORACLE.EXE (MMAN) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_mman_4888.trc |
| 10 | ORACLE.EXE (DBW0) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_dbw0_5264.trc |
| 11 | ORACLE.EXE (LGWR) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_lgwr_5304.trc |
| 12 | ORACLE.EXE (CKPT) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_ckpt_2100.trc |
| 13 | ORACLE.EXE (SMON) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_smon_5312.trc |
| 14 | ORACLE.EXE (RECO) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_reco_5588.trc |
| 15 | ORACLE.EXE (MMON) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_mmon_4712.trc |
| 16 | ORACLE.EXE (MMNL) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_mmnl_5608.trc |
| 17 | ORACLE.EXE (D000) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_d000_5688.trc |
| 18 | ORACLE.EXE (S000) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_s000_5692.trc |
| 19 | ORACLE.EXE (FBDA) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_fbda_5432.trc |
| 20 | ORACLE.EXE (SMCO) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_smco_4612.trc |
| 21 | ORACLE.EXE (VW000) | c:\oracle\diag\rdbms\ora11\ora11\trace\ora11_w000_3320.trc |



Bind Peeking and Adaptive Cursors

Bind Peeking + Adaptive Cursors

```
SELECT resource_code, count(*)  
FROM   bookings_large  
GROUP BY resource_code;
```

| RESOURCE_CODE | COUNT(*) |
|---------------|----------|
| VCR1 | 497009 |
| CONF | 497019 |
| LNCH | 745520 |
| BRSM | 745524 |
| PC1 | 48510 |
| FLPC | 497015 |
| BRLG | 1991051 |
| TAP1 | 248509 |
| VCR2 | 497011 |



Minority value



Majority value



Bind Peeking

Look at the value of the bind variable when the statement is parsed

Plan is based on that value

```
BEGIN :v3 := 'PC1'; END;
```

```
SELECT COUNT(quantity)  
FROM bookings_large  
WHERE resource_code = :v3;
```



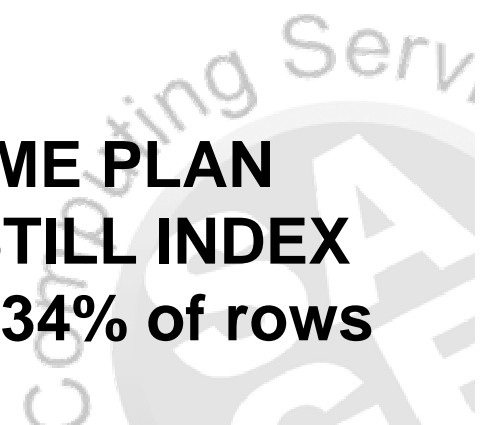
INDEX
for <1% of rows

```
BEGIN :v3 := 'BRLG'; END;
```

```
SELECT COUNT(quantity)  
FROM bookings_large  
WHERE resource_code = :v3;
```



SAME PLAN
→ STILL INDEX
for 34% of rows



Bind Peeking + Adaptive Cursors

```
SELECT COUNT(l.quantity) FROM train.bookings_large l  
WHERE resource_code = :v1
```

```
SELECT sql_id FROM v$sqlarea  
WHERE sql_text = 'SELECT COUNT(l.quantity)  
FROM train.bookings_large l WHERE resource_code = :v1'
```

| SQL_ID |
|---------------|
| 95jktg3mza0qm |

Bind Peeking + Adaptive Cursors

```
BEGIN :v1 := 'PC1'; END;  
SELECT COUNT(l.quantity) FROM train.bookings_large l  
WHERE resource_code = :v1  
  
SELECT sql_id, child_number, is_bind_sensitive, is_bind_aware  
FROM v$sql  
WHERE sql_id = '95jktg3mza0qm';
```

| SQL_ID | CHILD_NUMBER | IS_BIND_SENSITIVE | IS_BIND_AWARE |
|---------------|--------------|-------------------|---------------|
| 95jktg3mza0qm | 0 | Y | N |

IS_BIND_SENSITIVE - may need to change the plan
- set on first execution of the statement

IS_BIND_AWARE - do need to change the plan

Bind Peeking + Adaptive Cursors

```
SELECT child_number, bind_set_hash_value, peeked, executions,  
       rows_processed, buffer_gets  
FROM   v$sql_cs_statistics  
WHERE  sql_id = '95jktg3mza0qm'
```

| CHILD_NUMBER | BIND_SET_HASH_VALUE | PEEKED | EXECUTIONS | ROWS_PROCESSED | BUFFER_GETS |
|--------------|---------------------|--------|------------|----------------|-------------|
| 0 | 1466228028 | Y | 1 | 291062 | 6746 |

Bind Peeking + Adaptive Cursors

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|------|-------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.01 | 0.01 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 0.12 | 0.43 | 119 | 6635 | 0 | 1 |
| total | 4 | 0.14 | 0.44 | 119 | 6635 | 0 | 1 |

Misses in library cache during parse: 1

Misses in library cache during execute: 1

Optimizer mode: ALL_ROWS

Parsing user id: 82

Rows Row Source Operation

```

1  SORT AGGREGATE (cr=6635 pr=119 pw=119 time=0 us)
48510  TABLE ACCESS BY INDEX ROWID BOOKINGS_LARGE (cr=6635 pr=119 pw=119 time=2870 us
cost=1977 size=316008 card=39501)
48510  INDEX RANGE SCAN BK_RES2 (cr=104 pr=0 pw=0 time=527 us cost=90 size=0
card=39501)(object id 69870)

```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| SQL*Net message from client | 2 | 0.00 | 0.00 |

Bind Peeking + Adaptive Cursors

```
BEGIN :v1 := 'BRLG'; END;
```

```
SELECT COUNT(l.quantity) FROM train.bookings_large l
```

```
WHERE resource_code = :v1
```

```
SELECT sql_id, child_number, is_bind_sensitive, is_bind_aware
```

```
FROM v$sql
```

```
WHERE sql_id = '95jktg3mza0qm';
```

| SQL_ID | CHILD_NUMBER | IS_BIND_SENSITIVE | IS_BIND_AWARE |
|---------------|--------------|-------------------|---------------|
| 95jktg3mza0qm | 0 | Y | N |

Bind Peeking + Adaptive Cursors

```
SELECT child_number, bind_set_hash_value, peeked, executions,  
       rows_processed, buffer_gets  
FROM   v$sql_cs_statistics  
WHERE  sql_id = '95jktg3mza0qm'
```

| CHILD_NUMBER | BIND_SET_HASH_VALUE | PEEKED | EXECUTIONS | ROWS_PROCESSED | BUFFER_GETS |
|--------------|---------------------|--------|------------|----------------|-------------|
| 0 | 1466228028 | Y | 1 | 291062 | 6746 |

Bind Peeking + Adaptive Cursors

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|--------------|-------|--------------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 6.29 | 78.82 | 34335 | 39594 | 0 | 1 |
| total | 4 | 6.29 | 78.82 | 34335 | 39594 | 0 | 1 |

Misses in library cache during parse: 0

Optimizer mode: ALL_ROWS

Parsing user id: 82

Rows Row Source Operation

```

1  SORT AGGREGATE (cr=39594 pr=34335 pw=34335 time=0 us)
1991051  TABLE ACCESS BY INDEX ROWID BOOKINGS_LARGE (cr=39594 pr=34335 pw=34335 time=626800
        us cost=1977 size=316008 card=39501)
1991051  INDEX RANGE SCAN BK_RES2 (cr=4437 pr=4436 pw=4436 time=61616 us cost=90 size=0
        card=39501)(object id 69870)
  
```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| db file sequential read | 34335 | 0.29 | 72.08 |
| resmgr:cpu quantum | 30 | 0.10 | 1.29 |
| SQL*Net message from client | 2 | 0.00 | 0.00 |

Bind Peeking + Adaptive Cursors

```
BEGIN :v1 := 'BRLG'; END;  
SELECT COUNT(l.quantity) FROM train.bookings_large l  
WHERE resource_code = :v1
```

```
SELECT sql_id, child_number, is_bind_sensitive, is_bind_aware  
FROM v$sql  
WHERE sql_id = '95jktg3mza0qm';
```

| SQL_ID | CHILD_NUMBER | IS_BIND_SENSITIVE | IS_BIND_AWARE |
|---------------|--------------|-------------------|---------------|
| 95jktg3mza0qm | 0 | Y | N |
| 95jktg3mza0qm | 1 | Y | Y |

Bind Peeking + Adaptive Cursors

```
SELECT child_number, bind_set_hash_value, peeked, executions,  
       rows_processed, buffer_gets  
FROM   v$sql_cs_statistics  
WHERE  sql_id = '95jktg3mza0qm'
```

| CHILD_NUMBER | BIND_SET_HASH_VALUE | PEEKED | EXECUTIONS | ROWS_PROCESSED | BUFFER_GETS |
|--------------|---------------------|--------|------------|----------------|-------------|
| 1 | 2982103524 | Y | 1 | 3982104 | 35346 |
| 0 | 1466228028 | Y | 1 | 291062 | 6746 |

Bind Peeking + Adaptive Cursors

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|--------------|-------|--------------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 1.23 | 21.71 | 35341 | 35346 | 0 | 1 |
| total | 4 | 1.23 | 21.72 | 35341 | 35346 | 0 | 1 |

Misses in library cache during parse: 0

Misses in library cache during execute: 1

Optimizer mode: ALL_ROWS

Parsing user id: 82

Rows Row Source Operation

```

1  SORT AGGREGATE (cr=35346 pr=35341 pw=35341 time=0 us)
1991051  TABLE ACCESS FULL BOOKINGS_LARGE (cr=35346 pr=35341 pw=35341 time=414975 us
cost=9652 size=15700664 card=1962583)
  
```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| direct path read | 314 | 0.42 | 19.45 |
| SQL*Net message from client | 2 | 0.10 | 0.10 |

Bind Peeking + Adaptive Cursors - Executions

```
SELECT s.executions v$sqllex, c.sql_id, c.executions cs_stat_exec ,  
       c.peeked  
FROM v$sql_cs_statistics c, v$sql s  
WHERE s.sql_id = c.sql_id  
AND s.child_number = c.child_number  
AND s.sql_id = '95jktg3mza0qm'
```

|  V\$SQLEX |  SQL_ID |  CS_STAT_EXEC |  PEEKED |
|--|--|--|--|
| 1 | 95jktg3mza0qm | 1 | Y |
| 4 | 95jktg3mza0qm | 1 | Y |

Bind Peeking + Adaptive Cursors - Summary

Statements with histograms and bind variables are bind sensitive

The first time you execute a statement with different selectivity it uses the original plan

The second time it changes the plan and become bind aware

Be careful when statements become invalidated by:-

- Gathering statistics

- Flushing the shared pool

- Restarting the database

- Or when they are aged out

Otherwise you don't need to worry any more





Gathering Statistics

Multi Column Statistics

Gather statistics on combinations of columns

Improves plans for complex dependency patterns

```
SELECT count(b.comments)
FROM train.events_large e, train.bookings_large b
WHERE e.org_id = 2264
AND   e.event_no = b.event_no
AND   e.comments = 'TEST'
```



Multi Column Data Patterns

```
SELECT count(*)  
FROM train.events_large e
```

| A Z | COUNT(*) |
|-----|----------|
| | 100322 |

```
SELECT count(*)  
FROM train.events_large e  
WHERE e.org_id = 2264
```

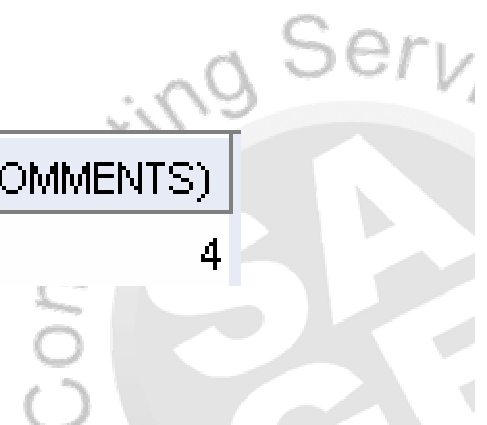
| A Z | COUNT(*) |
|-----|----------|
| | 25083 |

```
SELECT count(e.comments)  
FROM train.events_large e  
WHERE e.comments = 'TEST'
```

| A Z | COUNT(E.COMMENTS) |
|-----|-------------------|
| | 75243 |

```
SELECT count(e.comments)  
FROM train.events_large e  
WHERE e.comments = 'TEST'  
AND e.org_id = 2264
```

| A Z | COUNT(E.COMMENTS) |
|-----|-------------------|
| | 4 |



With Individual Statistics

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|-------|-------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 5.00 | 22.52 | 35341 | 36337 | 0 | 1 |
| total | 4 | 5.00 | 22.52 | 35341 | 36337 | 0 | 1 |

Misses in library cache during parse: 1

Optimizer mode: ALL_ROWS

Parsing user id: 82

| Rows | Row Source Operation |
|---------|--|
| 1 | SORT AGGREGATE (cr=36337 pr=35341 pw=35341 time=0 us) |
| 175 | HASH JOIN (cr=36337 pr=35341 pw=35341 time=13 us cost=9966 size=33365083 card=1076293) |
| 4 | TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=3 us cost=274 size=319770 card=18810) |
| 5767168 | TABLE ACCESS FULL BOOKINGS_LARGE (cr=35346 pr=35341 pw=35341 time=225583 us cost=9665 size=80740352 card=5767168) |

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| direct path read | 314 | 0.38 | 15.88 |
| resmgr:cpu quantum | 11 | 0.14 | 0.27 |
| SQL*Net message from client | 2 | 0.04 | 0.04 |






Auto Created Column Statistics

BEGIN

```
dbms_stats.gather_table_stats(ownname=>'TRAIN',  
    tabname=>'EVENTS_LARGE', cascade=>TRUE,  
    method_opt => 'FOR ALL COLUMNS SIZE AUTO',  
    no_invalidate=>FALSE);
```

END;

```
SELECT column_name, histogram, num_buckets  
FROM   user_tab_col_statistics  
WHERE  table_name = 'EVENTS_LARGE'
```

|  COLUMN_NAME |  HISTOGRAM |  NUM_BUCKETS |
|---|---|---|
| EVENT_NO | NONE | 1 |
| ORG_ID | FREQUENCY | 6 |
| DESCRIPTION | NONE | 1 |
| CONTACT_NAME | NONE | 1 |
| START_DATE | NONE | 1 |
| END_DATE | NONE | 1 |
| COMMENTS | FREQUENCY | 2 |

Auto Created Column Statistics

```
SELECT extension_name, extension
FROM user_stat_extensions
WHERE table_name='EVENTS_LARGE'
```

| EXTENSION_NAME | EXTENSION |
|----------------|-----------|
| | |

Same access path used

| Rows | Row | Source | Operation |
|---------|-----|--------|---|
| | 1 | | SORT AGGREGATE (cr=36337 pr=35341 pw=35341 time=0 us) |
| | 175 | | HASH JOIN (cr=36337 pr=35341 pw=35341 time=13 us cost=9966 size=33365083 card=1076293) |
| | 4 | | TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=3 us cost=274 size=319770 card=18810) |
| 5767168 | | | TABLE ACCESS FULL BOOKINGS_LARGE (cr=35346 pr=35341 pw=35341 time=225583 us cost=9665 size=80740352 card=5767168) |

Manually Create Multi Column Statistics

BEGIN

```
dbms_stats.gather_table_stats(ownname=>'TRAIN',  
    tabname=>'EVENTS_LARGE', cascade=>TRUE,  
    method_opt => 'FOR COLUMNS (org_id, comments) SIZE  
    SKEWONLY', no_invalidate=>FALSE);
```




END;

```
SELECT extension_name, extension  
FROM   user_stat_extensions  
WHERE  table_name='EVENTS_LARGE'
```

| A Z | EXTENSION_NAME | EXTENSION |
|-----|---------------------------------|-----------------------|
| | SYS_STUXRIN\$UJ2J7FKD#I4PEGV6BE | ("ORG_ID","COMMENTS") |

Manually Create Multi Column Statistics

```
SELECT column_name, histogram, num_buckets  
FROM   user_tab_col_statistics  
WHERE  table_name = 'EVENTS_LARGE'
```

|  COLUMN_NAME |  HISTOGRAM |  NUM_BUCKETS |
|---|---|---|
| EVENT_NO | NONE | 1 |
| ORG_ID | FREQUENCY | 6 |
| DESCRIPTION | NONE | 1 |
| CONTACT_NAME | NONE | 1 |
| START_DATE | NONE | 1 |
| END_DATE | NONE | 1 |
| COMMENTS | FREQUENCY | 2 |
| SYS_STUXRIN\$UJ... | FREQUENCY | 8 |

Multi Column Statistics

```
SELECT count(b.comments)
FROM train.events_large e, train.bookings_large b
WHERE e.org_id = 2264
AND    e.event_no = b.event_no
AND    e.comments = 'TEST'
```

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|------|-------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 0.01 | 0.01 | 0 | 1175 | 0 | 1 |
| total | 4 | 0.01 | 0.01 | 0 | 1175 | 0 | 1 |

Rows Row Source Operation

```
-----
1  SORT AGGREGATE (cr=1175 pr=0 pw=0 time=0 us)
175 NESTED LOOPS (cr=1175 pr=0 pw=0 time=71 us)
175 NESTED LOOPS (cr=1001 pr=0 pw=0 time=39 us cost=1129 size=26815 card=865)
4   TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=4 us cost=274 size=255 card=15)
175 INDEX RANGE SCAN BK_EVT2 (cr=10 pr=0 pw=0 time=10 us cost=2 size=0 card=57)(object id 69868)
175 TABLE ACCESS BY INDEX ROWID BOOKINGS_LARGE (cr=174 pr=0 pw=0 time=0 us cost=57 size=798 card=57)
```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| SQL*Net message from client | 2 | 0.02 | 0.02 |



Multi Column Statistics

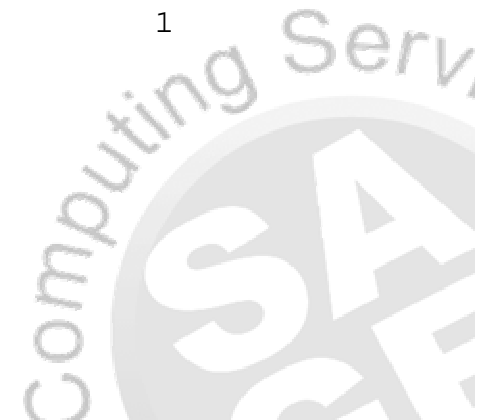
```
SELECT count(b.comments)
FROM train.organisations o, train.events_large e, train.bookings_large b,
     train.resources r
WHERE o.org_id = e.org_id
AND   e.event_no = b.event_no
AND   b.resource_code = r.code
AND   o.name = 'Australian Medical Systems'
AND   r.description = 'Buffet Lunch'
AND   e.comments = 'TEST'
```

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|-------|-------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 4.64 | 21.66 | 35341 | 36351 | 0 | 1 |
| total | 4 | 4.64 | 21.66 | 35341 | 36351 | 0 | 1 |

Misses in library cache during parse: 0

Optimizer mode: ALL_ROWS

Parsing user id: 82



Multi Column Statistics

Rows Row Source Operation

1 SORT AGGREGATE (cr=36351 pr=35341 pw=35341 time=0 us)

25 HASH JOIN (cr=36351 pr=35341 pw=35341 time=1966789 us cost=9972 size=6555490
card=79945)

4 HASH JOIN (cr=1005 pr=0 pw=0 time=8 us cost=281 size=792225 card=12575)

1 MERGE JOIN CARTESIAN (cr=14 pr=0 pw=0 time=0 us cost=6 size=46 card=1)

1 TABLE ACCESS FULL ORGANISATIONS (cr=7 pr=0 pw=0 time=0 us cost=3 size=26
card=1)

1 BUFFER SORT (cr=7 pr=0 pw=0 time=0 us cost=3 size=20 card=1)

1 TABLE ACCESS FULL RESOURCES (cr=7 pr=0 pw=0 time=0 us cost=3 size=20
card=1)

75243 TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=1020 us cost=274
size=1282616 card=75448)

5767168 **TABLE ACCESS FULL BOOKINGS_LARGE** (cr=35346 pr=35341 pw=35341 time=350849 us
cost=9665 size=109576192 card=5767168)

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| ----- | ----- | ----- | ----- |
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| direct path read | 314 | 0.35 | 15.49 |
| resmgr:cpu quantum | 2 | 0.12 | 0.12 |
| SQL*Net message from client | 2 | 0.04 | 0.05 |

Multi Column Statistics - Summary

Manually create column groups and statistics on them

Complex joins may still need SQL Profiles

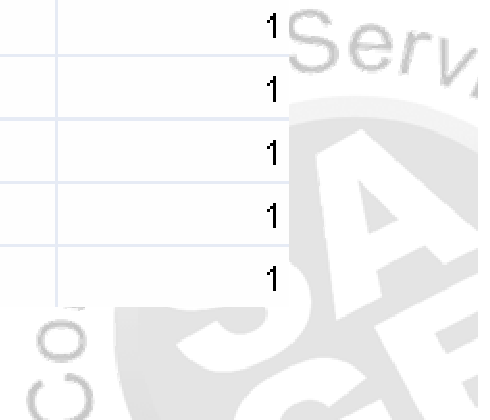
Use for dependent column such as Country, State



Expression Statistics

```
SELECT comments, count(*)  
FROM events_large  
GROUP by comments  
ORDER by count(*) desc
```

| COMMENTS | COUNT(*) |
|--|----------|
| TEST | 74492 |
| (null) | 16554 |
| This is the final meeting for this project | 8277 |
| Comments5 | 1 |
| Comments8 | 1 |
| Comments12 | 1 |
| Comments58 | 1 |
| Comments63 | 1 |
| Comments82 | 1 |
| Comments88 | 1 |
| Comments99 | 1 |
| Comments102 | 1 |
| Comments103 | 1 |
| Comments104 | 1 |
| Comments106 | 1 |
| Comments108 | 1 |






Expression Statistics

BEGIN

```
dbms_stats.gather_table_stats(ownname=>'TRAIN',  
    tabname=>'EVENTS_LARGE', cascade=>TRUE,  
    method_opt => 'FOR ALL COLUMNS SIZE AUTO',  
    no_invalidate=>FALSE);
```

END;

```
SELECT column_name, histogram, num_buckets  
FROM   user_tab_col_statistics  
WHERE  table_name = 'EVENTS_LARGE'
```

|  COLUMN_NAME |  HISTOGRAM |  NUM_BUCKETS |
|---|---|---|
| EVENT_NO | NONE | 1 |
| ORG_ID | FREQUENCY | 6 |
| DESCRIPTION | NONE | 1 |
| CONTACT_NAME | NONE | 1 |
| START_DATE | NONE | 1 |
| END_DATE | NONE | 1 |
| COMMENTS | HEIGHT BALAN... | 254 |
| SYS_STUXRIN\$UJ... | HEIGHT BALAN... | 254 |

Expression Statistics

```
SELECT count(b.comments)
FROM train.events_large e, train.bookings_large b
WHERE e.event_no = b.event_no
AND    e.comments = 'Comments8'
```

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|------|-------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 0.01 | 0.01 | 0 | 1052 | 0 | 1 |
| total | 4 | 0.01 | 0.01 | 0 | 1052 | 0 | 1 |

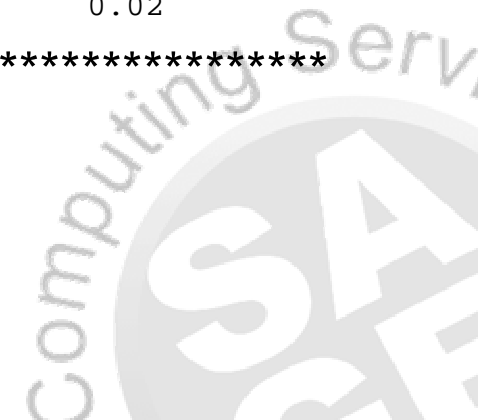


Expression Statistics

| Rows | Row Source Operation |
|------|--|
| 1 | SORT AGGREGATE (cr=1052 pr=0 pw=0 time=0 us) |
| 58 | NESTED LOOPS (cr=1052 pr=0 pw=0 time=28 us) |
| 58 | NESTED LOOPS (cr=994 pr=0 pw=0 time=7 us cost=331 size=1178 card=38) |
| 1 | TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=0 us cost=274 size=17 card=1) |
| 58 | INDEX RANGE SCAN BK_EVT2 (cr=3 pr=0 pw=0 time=3 us cost=2 size=0 card=57)(object id 69868) |
| 58 | TABLE ACCESS BY INDEX ROWID BOOKINGS_LARGE (cr=58 pr=0 pw=0 time=0 us cost=57 size=798 card=57) |

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| SQL*Net message from client | 2 | 0.02 | 0.02 |



Expression Statistics

```
SELECT count(b.comments)
FROM train.events_large e, train.bookings_large b
WHERE e.event_no = b.event_no
AND    initcap(e.comments) = 'Comments8'
```

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|-------|-------|---------|------|
| Parse | 1 | 0.01 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 5.70 | 21.52 | 35341 | 36337 | 0 | 1 |
| total | 4 | 5.71 | 21.52 | 35341 | 36337 | 0 | 1 |

Expression Statistics

```

Rows      Row Source Operation
-----
      1  SORT AGGREGATE (cr=36337 pr=35341 pw=35341 time=0 us)
     58  HASH JOIN  (cr=36337 pr=35341 pw=35341 time=624644 us cost=9967 size=1779493
card=57403)
      1  TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=0 us cost=275
size=17051 card=1003)
5767168  TABLE ACCESS FULL BOOKINGS_LARGE (cr=35346 pr=35341 pw=35341 time=301177 us
cost=9665 size=80740352 card=5767168)
  
```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| direct path read | 314 | 0.41 | 15.12 |
| resmgr:cpu quantum | 2 | 0.10 | 0.10 |
| SQL Net message from client | 2 | 0.03 | |



Expression Statistics

BEGIN

```
dbms_stats.gather_table_stats(ownname=>'TRAIN',  
    tabname=>'EVENTS_LARGE', cascade=>TRUE,  
    method_opt => 'FOR COLUMNS (INITCAP(comments)) SIZE  
    SKEWONLY', no_invalidate=>FALSE);
```




END;

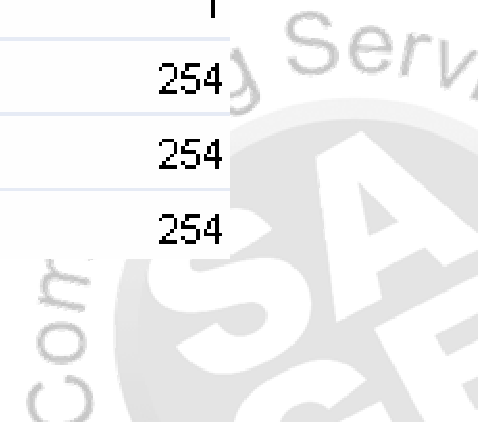
```
SELECT extension_name, extension  
FROM   user_stat_extensions  
WHERE  table_name='EVENTS_LARGE'
```

| EXTENSION_NAME | EXTENSION |
|---------------------------------|-----------------------|
| SYS_STUXRIN\$UJ2J7FKD#I4PEGV6BE | ("ORG_ID","COMMENTS") |
| SYS_STU7YKZITPFB#\$EAK3UP7_23FY | (INITCAP("COMMENTS")) |

Expression Statistics

```
SELECT column_name, histogram, num_buckets
FROM   user_tab_col_statistics
WHERE  table_name = 'EVENTS_LARGE'
```

|  COLUMN_NAME |  HISTOGRAM |  NUM_BUCKETS |
|---|---|---|
| EVENT_NO | NONE | 1 |
| ORG_ID | FREQUENCY | 6 |
| DESCRIPTION | NONE | 1 |
| CONTACT_NAME | NONE | 1 |
| START_DATE | NONE | 1 |
| END_DATE | NONE | 1 |
| COMMENTS | HEIGHT BALAN... | 254 |
| SYS_STUXRIN\$UJ2J7FKD#I4PEGV6BE | HEIGHT BALAN... | 254 |
| SYS_STU7YKZITPFB#\$EAK3UP7_23FY | HEIGHT BALAN... | 254 |



Expression Statistics

```
SELECT count(b.comments)
FROM train.events_large e, train.bookings_large b
WHERE e.event_no = b.event_no
AND    initcap(e.comments) = 'Comments8'
```

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|------|-------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 2 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 2 | 0.03 | 0.04 | 0 | 1052 | 0 | 1 |
| total | 4 | 0.03 | 0.04 | 0 | 1054 | 0 | 1 |



Expression Statistics

| Rows | Row Source Operation |
|------|--|
| 1 | SORT AGGREGATE (cr=1052 pr=0 pw=0 time=0 us) |
| 58 | NESTED LOOPS (cr=1052 pr=0 pw=0 time=27 us) |
| 58 | NESTED LOOPS (cr=994 pr=0 pw=0 time=7 us cost=332 size=2223 card=57) |
| 1 | TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=0 us cost=275 size=25 card=1) |
| 58 | INDEX RANGE SCAN BK_EVT2 (cr=3 pr=0 pw=0 time=3 us cost=2 size=0 card=57)(object id 69868) |
| 58 | TABLE ACCESS BY INDEX ROWID BOOKINGS_LARGE (cr=58 pr=0 pw=0 time=0 us cost=57 size=798 card=57) |

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| SQL*Net message from client | 2 | 0.02 | 0.02 |





Gathering Statistics - Preferences

Setting Statistics Gathering Defaults

Obsolete:

GET_PARAM

SET_PARAM Procedure

Use:

GET_PREFS Function

SET_TABLE_PREFS (set for individual tables)

SET_DATABASE_PREFS (sets for all tables, can include/exclude
SYS tables)

SET_GLOBAL_PREFS – for new objects



Database Instance: ora11.sagecomputing.com.au > Tables > Manage Optimizer Statistics >

Logged in As SYS

Global Statistics Gathering Options

Database ora11.sagecomputing.com.au

Cancel

Show SQL

Apply

Statistics History

Retention Period
(days)

31

The number of days for which optimizer statistics history will be retained.

Gather Optimizer Statistics Default Options

Oracle recommends that you use the Gather Auto choice for the Gather Objects options when you use the Gather Optimizer Statistics process for Database and Schemas. If you choose not to use Gather Auto, the defaults for the other options are set here. Changing the options will impact the automated Optimizer Statistics Gathering task and user defined jobs.

Reset Defaults

Estimate Percentage



Auto (Oracle recommended)



100%



Percentage

Degree of Parallelism



Table default



Auto



System default



Degree

Granularity

Auto



Cursor Invalidation



Auto (Oracle recommended)



Immediate



None

Cascade



Auto (Oracle recommended)



True



False

Target Object Class (Auto Job)



Auto (Oracle recommended)



All



Oracle

Stale Percentage

10

Incremental



True



False

Publish



True



False

Histograms

FOR ALL COLUMNS SIZE AUTO



Database Instance: ora11.sagecomputing.com.au > Object Level Statistics Gathering Preferences >

Logged in As SYS

Edit Schema Preferences

Show SQL Cancel OK

Select schema and options. The options will be applied to all the tables under the selected schema.

Schema 

Estimate Percentage ☒ No Change ☐ Inherit Global ☐ Auto (Oracle recommended) ☐ 100% ☐ Percentage

Degree of Parallelism ☒ No Change ☐ Inherit Global ☐ Table default ☐ Auto ☐ System default ☐ Degree

Granularity ▼

Cursor Invalidation ☒ No Change ☐ Inherit Global ☐ Auto (Oracle recommended) ☐ Immediate ☐ None

Cascade ☒ No Change ☐ Inherit Global ☐ Auto (Oracle recommended) ☐ True ☐ False

Stale Percentage ☒ No Change ☐ Inherit Global ☐ Percentage

Incremental ☒ No Change ☐ Inherit Global ☐ True ☐ False

Publish ☒ No Change ☐ Inherit Global ☐ True ☐ False

Histograms ☒ No Change ☐ Inherit Global ☐ FOR ALL COLUMNS SIZE 1

Show SQL Cancel OK

Database | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owner

[About Oracle Enterprise Manager](#)

Database Instance: ora11.sagecomputing.com.au > Object Level Statistics Gathering Preferences >

Logged in As SYS

Add Table Preferences

Show SQL

Cancel

OK

General

Statistics Extension

* Table Name



Estimate Percentage ☒ Inherit Global ☐ Auto (Oracle recommended) ☐ 100% ☐ Percentage

Degree of Parallelism ☒ Inherit Global ☐ Table default ☐ Auto ☐ System default ☐ Degree

Granularity

Cursor Invalidation ☒ Inherit Global ☐ Auto (Oracle recommended) ☐ Immediate ☐ None

Cascade ☒ Inherit Global ☐ Auto (Oracle recommended) ☐ True ☐ False

Stale Percentage ☒ Inherit Global ☐ Percentage

Incremental ☒ Inherit Global ☐ True ☐ False

Publish ☒ Inherit Global ☐ True ☐ False

Histograms ☒ Inherit Global ☐ FOR COLUMNS (org_id, comments)
SIZE SKEWONLY



Doesn't work

General

Statistics Extension

Show SQL

Cancel

OK

Database | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Table Prefs

PROCEDURE SET_TABLE_PREFS

| Argument Name | Type | In/Out | Default? |
|---------------|----------|--------|----------|
| ----- | ----- | ----- | ----- |
| OWNNAME | VARCHAR2 | IN | |
| TABNAME | VARCHAR2 | IN | |
| PNAME | VARCHAR2 | IN | |
| PVALUE | VARCHAR2 | IN | |

PNAME

CASCADE

DEGREE

ESTIMATE_PERCENT

METHOD_OPT

NO_INVALIDATE

GRANULARITY

PUBLISH

INCREMENTAL (only scan partitions that have been changed)

STALE_PERCENT



Table Prefs - Example

BEGIN

```
dbms_stats.set_table_prefs(ownname=>'TRAIN',  
    tabname=>'EVENTS_LARGE',  
    pname=>'STALE_PERCENT', pvalue=>'30');
```

```
dbms_stats.set_table_prefs(ownname=>'TRAIN',  
    tabname=>'EVENTS_LARGE',  
    pname=>'METHOD_OPT',  
    pvalue=>'FOR ALL COLUMNS SIZE AUTO');
```

END;

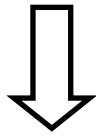
SELECT *

FROM DBA_TAB_STAT_PREFS;

| OWNER | TABLE_NAME | PREFERENCE_NAME | PREFERENCE_VALUE |
|-------|--------------|-----------------|---------------------------|
| TRAIN | EVENTS_LARGE | METHOD_OPT | FOR ALL COLUMNS SIZE AUTO |
| TRAIN | EVENTS_LARGE | STALE_PERCENT | 30 |

Gathering and Publishing

Set preferences to PUBLISH = 'FALSE'



Gather Statistics



user_tab_pending_stats
user_ind_pending_stats
user_col_pending_stats

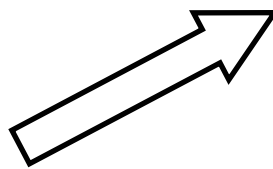
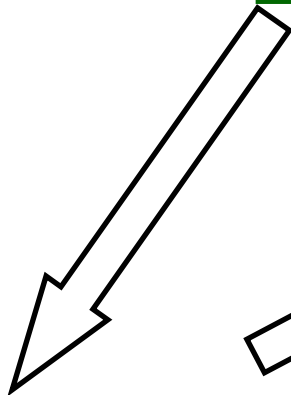
user_tab_statistics
user_ind_statistics
user_tab_col_statistics

Run test case

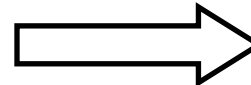
Alter session set

optimizer_pending_statistics
= TRUE

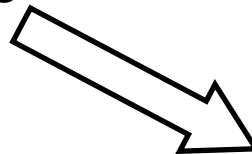
Run test case



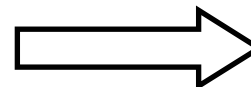
Better



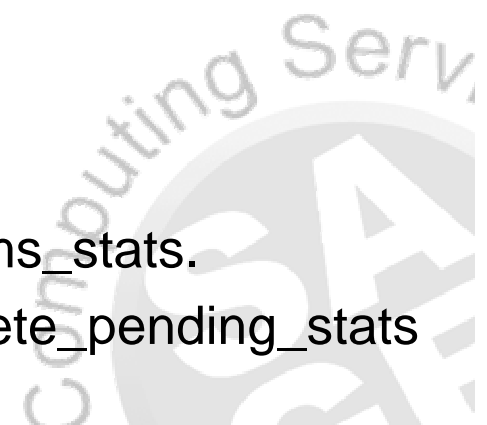
dbms_stats.
publish_pending_stats



Worse

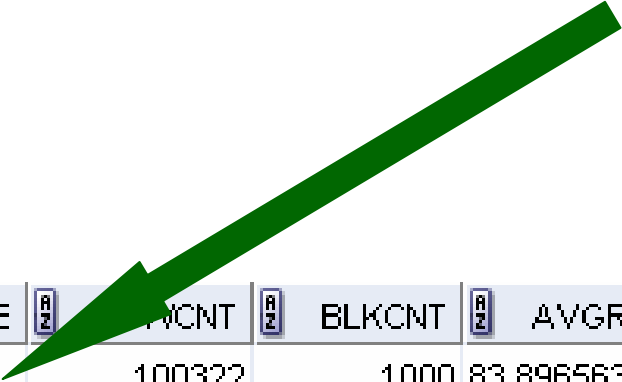


dbms_stats.
delete_pending_stats



Gathering and Publishing

```
SELECT obj#, TO_CHAR(savtime,'dd/mm/yyyy') save_time,  
       rowcnt, blkcnt, avgrln, samplesize, analyzetime  
FROM   wri$_optstat_tab_history  
ORDER BY savtime desc
```



| OBJ# | SAVE_TIME | ROWCNT | BLKCNT | AVGRLN | SAMPLESIZE | ANALYZETIME |
|-------|------------|--------|--------|----------------|------------|-------------|
| 69846 | 01/12/3000 | 100322 | 1000 | 83.89656306... | 100322 | 04/NOV/07 |
| 69670 | 04/11/2007 | 19344 | 244 | 46 | 19344 | 04/NOV/07 |
| 69656 | 04/11/2007 | 3833 | 118 | 230 | 3833 | 04/NOV/07 |
| 69742 | 04/11/2007 | 22122 | 95 | 26 | 22122 | 04/NOV/07 |

Gathering and Publishing

Never publish any statistics until you are happy they work better





Automatic Tuning

Automatic Tuning

Automatic SQL Tuning
(maintenance window)

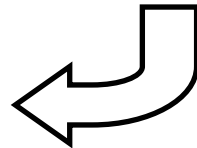


Identify high load candidates for tuning

Automatic Workload
Repository (AWR)

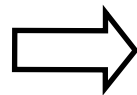
SQL Profiles

for each statement



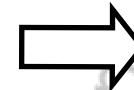
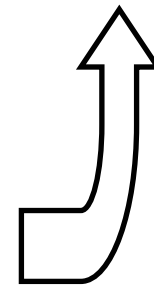
Run SQL Tuning
Advisor

Test with and
without profile



ACCEPT_SQL_PROFILES=true

ACCEPT_SQL_PROFILES=false
(default)



Report

Database Instance: ora11.sagecomputing.com.au >

Logged in As SYS

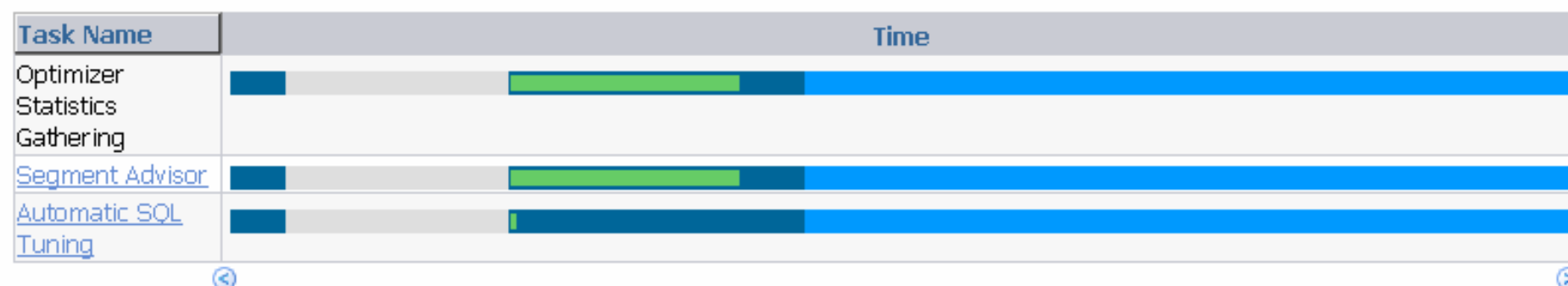
Automated Maintenance Tasks

Status **Enabled** [Configure](#)

Collected from Target 4/11/2007 11:15:36 AM WST

☒ **TIP** If the status is Disabled, there are no future windows.

* Begin Date  Interval
(example: 4/11/2007)



0:00 2 4 6 8 10 12 14 16 18 20 22
4/11/2007

Status Legend  Executed Task  Past Window  Future Window

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

[About Oracle Enterprise Manager](#)

Database Instance: ora11.sagecomputing.com.au > Automated Maintenance Tasks >

Logged in As SYS

[Show SQL](#) [Revert](#) [Apply](#)

Automated Maintenance Tasks Configuration

Global Status ☒ Enabled ☐ Disabled

Task Settings

Optimizer Statistics Gathering ☒ Enabled ☐ Disabled [Configure](#)

Segment Advisor ☒ Enabled ☐ Disabled

Automatic SQL Tuning ☒ Enabled ☐ Disabled [Configure](#)

Maintenance Window Group Assignment

[Edit Window Group](#)

| Window | Optimizer Statistics Gathering | Segment Advisor | Automatic SQL Tuning |
|----------------------------------|--|--|--|
| | Select All Select None | Select All Select None | Select All Select None |
| MONDAY WINDOW | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| TUESDAY WINDOW | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| WEDNESDAY WINDOW | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| THURSDAY WINDOW | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| FRIDAY WINDOW | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| SATURDAY WINDOW | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| SUNDAY WINDOW | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

[Show SQL](#) [Revert](#) [Apply](#)

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Database Instance: ora11.sagecomputing.com.au > Automated Maintenance Tasks Configuration >

Logged in As SYS

Automatic SQL Tuning Settings

[Show SQL](#) [Revert](#) [Apply](#)

Maximum Time Spent Per SQL During Tuning (sec)

Automatic Implementation of SQL Profiles ☒ Yes ☐ No

Maximum SQL Profiles Implemented Per Execution

Maximum SQL Profiles Implemented (Overall)

 **TIP** You need to login as SYS to make the change.

[Show SQL](#) [Revert](#) [Apply](#)

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
[About Oracle Enterprise Manager](#)

Database Instance: ora11.sagecomputing.com.au > Automated Maintenance Tasks >

Logged in As SYS

Automatic SQL Tuning Result Summary

The Automatic SQL Tuning runs during system maintenance windows as an automated maintenance task, searching for ways to improve the execution plans of high-load SQL statements.

Task Status

Automatic SQL Tuning (SYS_AUTO_SQL_TUNING_TASK) is currently **Enabled** [Configure](#)

Automatic Implementation of SQL Profiles is currently **Enabled**

Highly Recommended SQL Profiles **0**

Task Activity Summary

The activity summary graph shows the benefit of the task activities on the systems high-load SQL. Only profiles that significantly improve SQL performance were implemented.

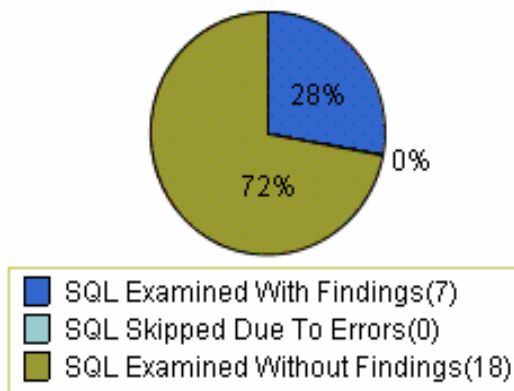
Time Period [Last 24 Hours](#) [Go](#) [View Report](#)

Begin Date 3/11/2007 13:10:13 (UTC+08:00) End Date 4/11/2007 13:10:13 (UTC+08:00)

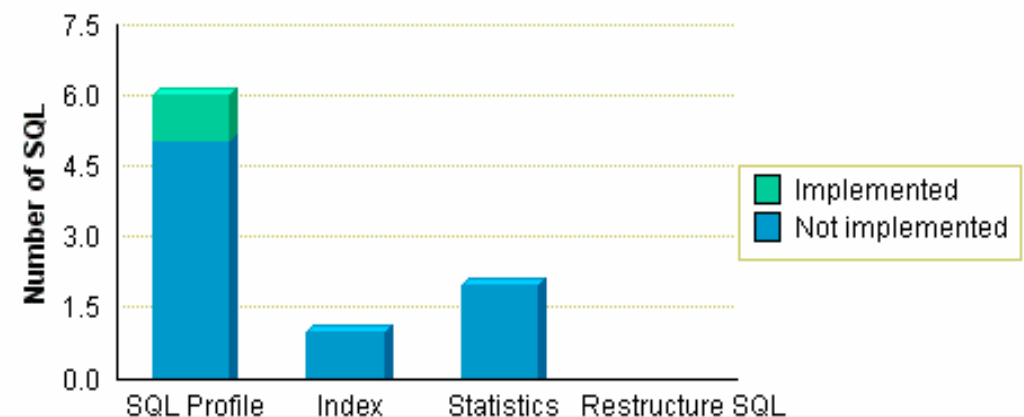
Overall Task Statistics

Executions **3** Candidate SQL **68** Distinct SQL Examined **25**

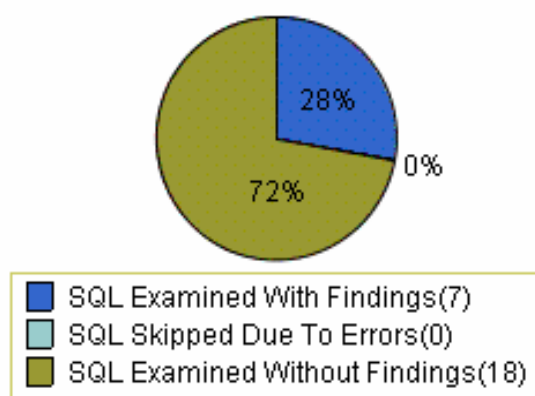
SQL Examined Status



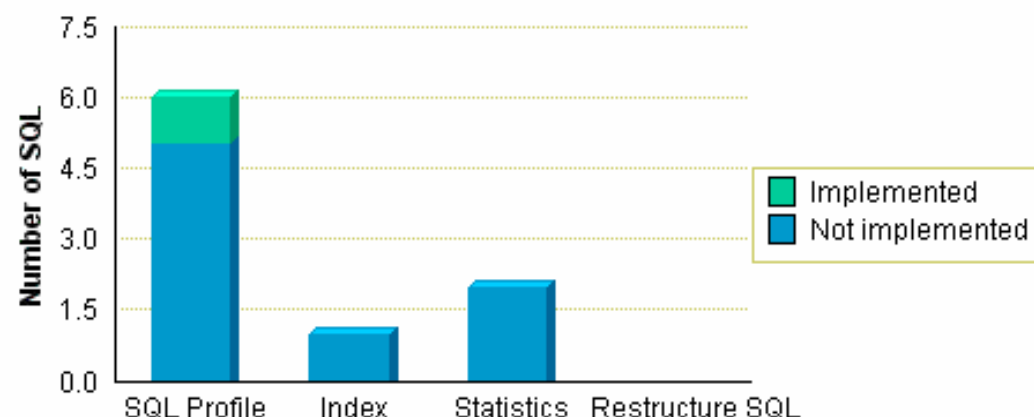
Breakdown by Finding Type



SQL Examined Status



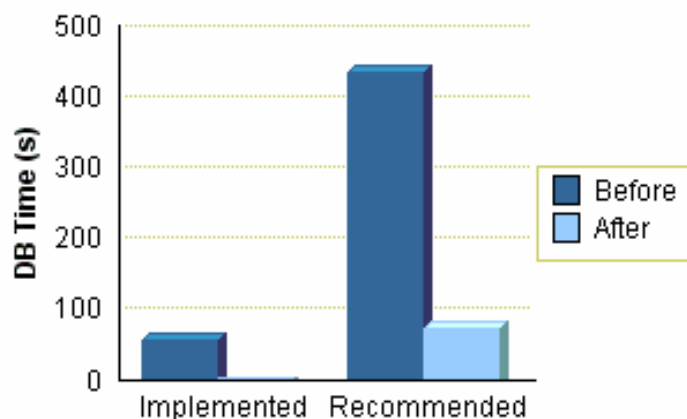
Breakdown by Finding Type



Profile Effect Statistics

Tuned SQL DB Time Benefit (seconds per week)

Implemented (sec) **52** Potential (sec) **362**



[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Database Instance: ora11.sagecomputing.com.au > Automated Maintenance Tasks > Automatic SQL Tuning > Logged in As SYS

Automatic SQL Tuning Result Details

Begin Date 3/11/2007 13:10:13 (UTC+08:00)

End Date 4/11/2007 13:10:13 (UTC+08:00)

Recommendations

Only profiles that significantly improve SQL performance were implemented.

View Recommendations

Implement All

Previous

1-25 of 68

Next 25

| Select | SQL Text | Parsing Schema | SQL ID | Statistics | SQL Profile | Index | Restructure SQL | Miscellaneous | Error | Date |
|----------------------------------|--|----------------|-------------------------------|------------|-------------|-------|-----------------|---------------|-------|--------|
| <input checked="" type="radio"/> | SELECT count(b.comments) FROM events_la... | TRAIN | cwvkrk0k6wv8 | | (96.5%) ✓ | | | | | 4/11/C |
| <input type="radio"/> | SELECT count(b.comments) FROM train.orga... | TRAIN | c84dzqxjkhxj4 | | (94.4%) ✓ | | | | | 4/11/C |
| <input type="radio"/> | SELECT :B1 TASK_ID, F.FINDING_ID FINDING... | DBSNMP | a8j39qb13tqkr | | (63.8%) ✓ | | | ✓ | | 4/11/C |
| <input type="radio"/> | select * from (SELECT OBJECT_NAME, OBJE... | TRAIN | 63m5m7ptp4wqd | ✓ | (63.2%) ✓ | | | | | 4/11/C |
| <input type="radio"/> | SELECT /*+ NO_MERGE(t) USE_NL(t) USE_NL(... | DBSNMP | 5qmzpd3hbxur | | (59.2%) ✓ | | | ✓ | | 4/11/C |
| <input type="radio"/> | with pri_cols as (SELECT cols.column_na... | SYS | 8904a0w3zpnkt | ✓ | (38.2%) ✓ | | | | | 4/11/C |
| <input type="radio"/> | SELECT count(b.comments) FROM train.even... | TRAIN | 5y60ynvxy2sk | | | | | ✓ | | 4/11/C |
| <input type="radio"/> | /* OracleOEM */ select 'table_sp... | DBSNMP | 49rr72c9uuyyq | | | | | ✓ | | 4/11/C |
| <input type="radio"/> | SELECT SUM(quantity) FROM train.bookings... | TRAIN | bjq1dhfgvtkhd | | | | | ✓ | | 4/11/C |
| <input type="radio"/> | SELECT count(b.comments) FROM events_la... | TRAIN | cwvkrk0k6wv8 | | | | | ✓ | | 4/11/C |
| <input type="radio"/> | /* OracleOEM */ SELECT PROPAGATION_NAM... | DBSNMP | bvf3fxv3hatw7 | | | | | ✓ | | 4/11/C |
| <input type="radio"/> | SELECT INSTANTIABLE, ... | SYSMAN | 6129566gyvx21 | | | | | ✓ | | 4/11/C |

Automatic Tuning

```
SELECT count(b.comments)
FROM train.organisations o, train.events_large e,
      train.bookings_large b, train.resources r
WHERE o.org_id = e.org_id
AND   e.event_no = b.event_no
AND   b.resource_code = r.code
AND   o.name = 'Australian Medical Systems'
AND   r.description = 'Buffet Lunch'
AND   e.comments = 'TEST'
```

Database Instance: ora11.sagecomputing.com.au > [Automated Maintenance Tasks](#) > [Automatic SQL Tuning](#) >
[Automatic SQL Tuning Result Details](#) > [Recommendations for SQL ID:c84dzqxjkhxj4](#) >

Logged in As SYS

Original Explain Plan (Annotated)

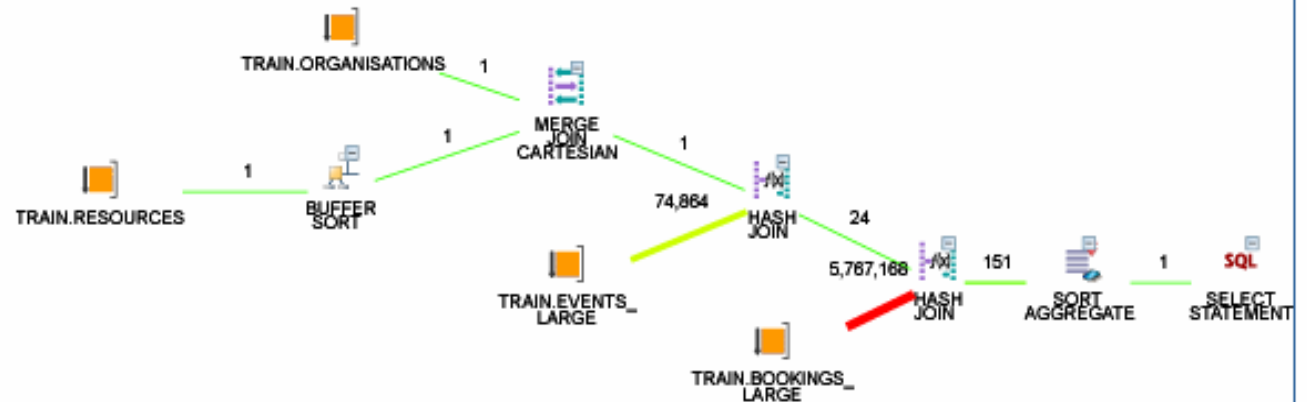
View ☒ Graph ☐ Table

Overview



Selection Details

Nothing Selected



TIP For an explanation of the icons and symbols used in this page, see the [Icon Key](#)

Database Instance: ora11.sagecomputing.com.au > Automated Maintenance Tasks > Automatic SQL Tuning > Automatic SQL Tuning Result Details > Recommendations for SQL ID:c84dzqxjkhxj4 >

Logged in As SYS

New Explain Plan With SQL Profile

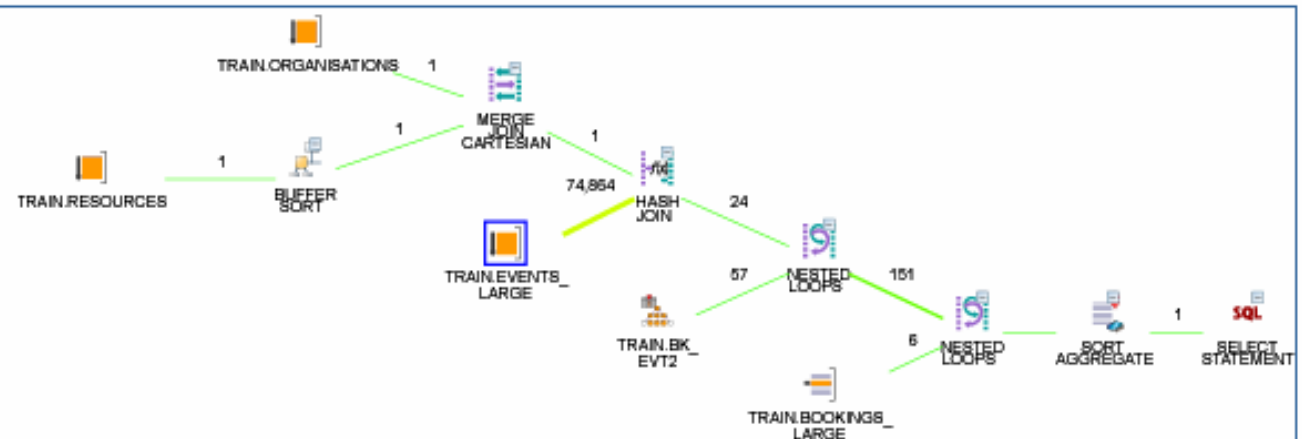
View ☒ Graph ☐ Table

Overview



Selection Details

Operation **TABLE ACCESS FULL**
Line ID **9**
Object [TRAIN.EVENTS_LARGE](#)
Object Type **TABLE**
Order **5**
Rows **74,864**
Size (KB) **1,242.859**
Cost **274**
Time **4**
CPU Cost **39,224,480**
I/O Cost **272**



TIP For an explanation of the icons and symbols used in this page, see the [Icon Key](#)

Automatic Tuning

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|-------|-------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 1 | 5.84 | 19.13 | 35341 | 36351 | 0 | 1 |
| total | 3 | 5.84 | 19.13 | 35341 | 36351 | 0 | 1 |

Rows Row Source Operation

```

1  SORT AGGREGATE (cr=36351 pr=35341 pw=35341 time=0 us)
149 HASH JOIN (cr=36351 pr=35341 pw=35341 time=66 us cost=9972 size=6504732 card=79326)
19  HASH JOIN (cr=1005 pr=0 pw=0 time=10 us cost=281 size=786051 card=12477)
1   MERGE JOIN CARTESIAN (cr=14 pr=0 pw=0 time=0 us cost=6 size=46 card=1)
1   TABLE ACCESS FULL ORGANISATIONS (cr=7 pr=0 pw=0 time=0 us cost=3 size=26 card=1)
1   BUFFER SORT (cr=7 pr=0 pw=0 time=0 us cost=3 size=20 card=1)
1   TABLE ACCESS FULL RESOURCES (cr=7 pr=0 pw=0 time=0 us cost=3 size=20 card=1)
74511 TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=1410 us cost=274 size=1272688
card=74864)
5767168 TABLE ACCESS FULL BOOKINGS_LARGE (cr=35346 pr=35341 pw=35341 time=233788 us
cost=9665 size=109576192 card=5767168)

```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| SQL*Net message from client | 2 | 4.20 | 4.20 |
| direct path read | 314 | 0.45 | 11.84 |



Automatic Tuning

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|------|-------|---------|------|
| Parse | 1 | 0.01 | 0.05 | 1 | 3 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 1 | 0.17 | 1.61 | 100 | 2078 | 0 | 1 |
| total | 3 | 0.18 | 1.66 | 101 | 2081 | 0 | 1 |

Rows Row Source Operation

```

1  SORT AGGREGATE (cr=2078 pr=100 pw=100 time=0 us)
149  NESTED LOOPS (cr=2078 pr=100 pw=100 time=137634 us)
1032  NESTED LOOPS (cr=1047 pr=14 pw=14 time=159 us cost=1649 size=12382 card=151)
19  HASH JOIN (cr=1005 pr=0 pw=0 time=19 us cost=281 size=1512 card=24)
1  MERGE JOIN CARTESIAN (cr=14 pr=0 pw=0 time=0 us cost=6 size=46 card=1)
1  TABLE ACCESS FULL ORGANISATIONS (cr=7 pr=0 pw=0 time=0 us cost=3 size=26 card=1)
1  BUFFER SORT (cr=7 pr=0 pw=0 time=0 us cost=3 size=20 card=1)
1  TABLE ACCESS FULL RESOURCES (cr=7 pr=0 pw=0 time=0 us cost=3 size=20 card=1)
74511  TABLE ACCESS FULL EVENTS_LARGE (cr=991 pr=0 pw=0 time=1127 us cost=274
size=1272688 card=74864)
1032  INDEX RANGE SCAN BK_EVT2 (cr=42 pr=14 pw=14 time=68 us cost=2 size=0
card=57)(object id 69868)
149  TABLE ACCESS BY INDEX ROWID BOOKINGS_LARGE (cr=1031 pr=86 pw=86 time=0 us cost=57
size=114 card=6)

```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| SQL*Net message from client | 2 | 17.58 | 17.58 |
| db file sequential read | 79 | 0.07 | 1.44 |

Automatic Tuning

```
SELECT signature, inuse_features, flags
FROM sql$
WHERE signature = '1697466342721355938'
```

| SIGNATURE | INUSE_FEATURES | FLAGS |
|---------------------|----------------|-------|
| 1697466342721355938 | 1 | 0 |

```
SELECT /*+ INDEX(sqlobj$ (signature category obj_type plan_id)) */
DISTINCT category
FROM sqlobj$ WHERE signature = '1697466342721355938'
```

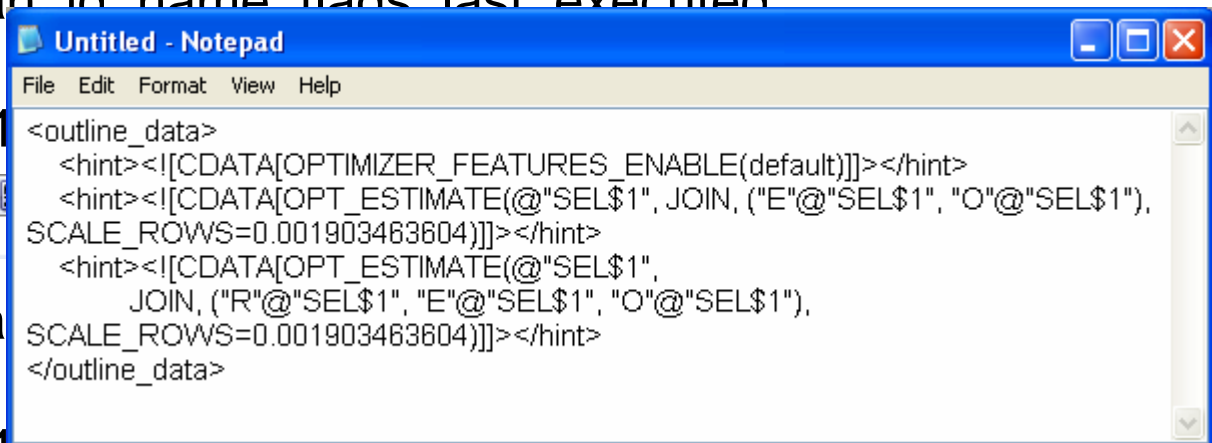
| CATEGORY |
|----------|
| DEFAULT |

```
SELECT obj_type, plan_id, name, flags, last_executed
FROM sqlobj$
WHERE signature = '1697466342721355938'
```

| OBJ_TYPE |
|----------|
| 1 |

```
SELECT obj_type, plan_id, comp_data
FROM sqlobj$data
WHERE signature = '1697466342721355938' AND category = .2
```

| OBJ_TYPE | PLAN_ID | COMP_DATA |
|----------|---------|--|
| 1 | 0 | <outline_data><hint><![CDATA[OPTIMIZER_FEATURES_ENABLE(default)]]></hint><hint><![CDATA[OPT_ESTIMATE(@SEL\$1, JOIN, ("E"@SEL\$1, "O"@SEL\$1), SCALE_ROWS=0.001903463604)]]></hint><hint><![CDATA[OPT_ESTIMATE(@SEL\$1, JOIN, ("R"@SEL\$1, "E"@SEL\$1, "O"@SEL\$1), SCALE_ROWS=0.001903463604)]]></hint></outline_data> |



Database Instance: ora11.sagecomputing.com.au > Automated Maintenance Tasks > Automatic SQL Tuning > Automatic SQL Tuning Result Details >

Logged in As SYS

Recommendations for SQL ID:cwxkrk0k6wtv8

[Return](#)

Only one recommendation should be implemented.

SQL Text

[SELECT count\(b.comments\) FROM events_large e, TRAIN.bookings_large b WHERE e.event_no = b.event_no AND e.event_no < 20000 AND b.resource_code = 'BRLG'](#)

Select Recommendation

[Original Explain Plan \(Annotated\)](#)

[Implement](#)

| Select | Type | Findings | Recommendations | Rationale | Benefit (%) | New Explain Plan | Compare Explain Plans |
|--------|-------------|---|---|-----------|-------------|------------------|-----------------------|
| | SQL Profile | A potentially better execution plan was found for this statement. | Consider accepting the recommended SQL profile. | | 96.54 | | |

[Return](#)

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

[About Oracle Enterprise Manager](#)

Select the plan hash value to see the details below. Plan Hash Value

[Statistics](#)

[Activity](#)

Plan

[Plan Control](#)

[Tuning History](#)

Data Source [Snapshot \(45\)](#)

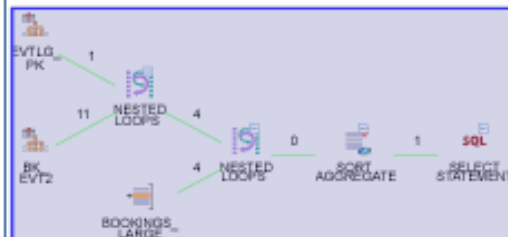
Capture Time **4/11/2007 11:00:41 (UTC+08:00)**

Parsing Schema **TRAIN** Optimizer Mode **ALL_ROWS**

Additional Information

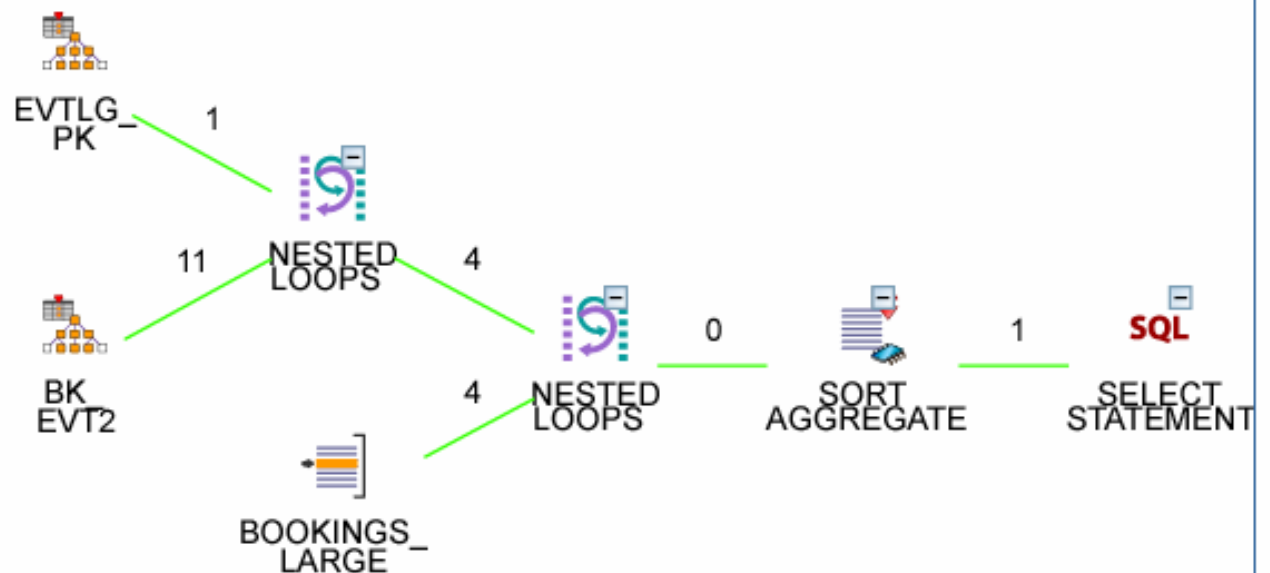
View ☒ Graph ☐ Table

Overview



Selection Details

Nothing Selected



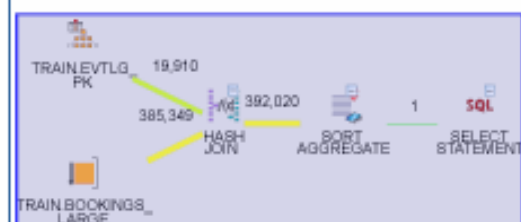
Database Instance: ora11.sagecomputing.com.au > Automated Maintenance Tasks > Automatic SQL Tuning > Automatic SQL Tuning Result Details > Recommendations for SQL ID:cwxkrk0k6wtv8 >

Logged in As SYS

New Explain Plan With SQL Profile

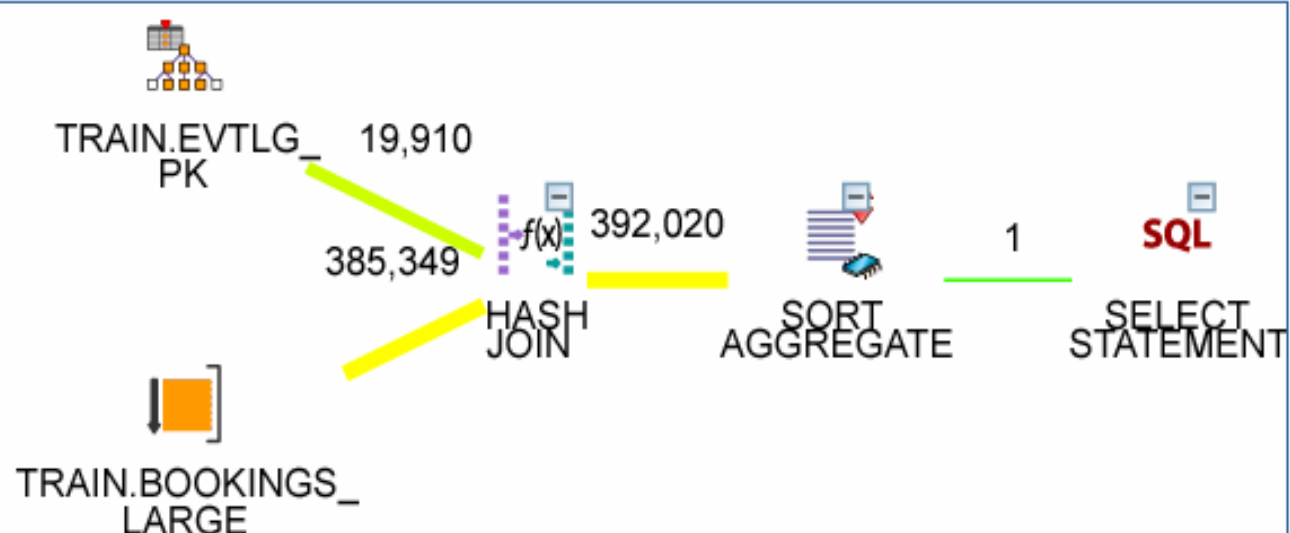
View ☒ Graph ☐ Table

Overview



Selection Details

Nothing Selected



TIP For an explanation of the icons and symbols used in this page, see the [Icon Key](#)

Automatic Tuning

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|-------|---------------|------|---------|---------|------|
| Parse | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 1 | 17.67 | 227.31 | 9818 | 1177977 | 0 | 1 |
| total | 3 | 17.67 | 227.32 | 9818 | 1177977 | 0 | 1 |

Rows Row Source Operation

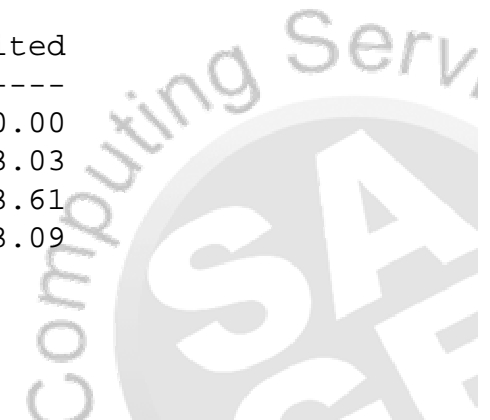
```

1  SORT AGGREGATE (cr=1177977 pr=9818 pw=9818 time=0 us)
398155  NESTED LOOPS (cr=1177977 pr=9818 pw=9818 time=1802295 us)
1148142  NESTED LOOPS (cr=29836 pr=2508 pw=2508 time=209241 us cost=15 size=144 card=4)
19800    INDEX RANGE SCAN EVTLG_PK (cr=37 pr=37 pw=37 time=395 us cost=2 size=17
card=1)(object id 69847)
1148142  INDEX RANGE SCAN BK_EVT2 (cr=29799 pr=2471 pw=2471 time=545049 us cost=2 size=0
card=11)(object id 69868)
398155  TABLE ACCESS BY INDEX ROWID BOOKINGS_LARGE (cr=1148141 pr=7310 pw=7310 time=0 us
cost=13 size=76 card=4)

```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| SQL*Net message from client | 2 | 283.03 | 283.03 |
| db file sequential read | 7621 | 0.71 | 193.61 |
| db file scattered read | 496 | 0.60 | 13.09 |



Automatic Tuning

| call | count | cpu | elapsed | disk | query | current | rows |
|---------|-------|------|---------|-------|-------|---------|------|
| Parse | 1 | 0.01 | 0.03 | 1 | 3 | 0 | 0 |
| Execute | 1 | 0.00 | 0.00 | 0 | 0 | 0 | 0 |
| Fetch | 1 | 1.48 | 18.43 | 35341 | 35383 | 0 | 1 |
| total | 3 | 1.50 | 18.47 | 35342 | 35386 | 0 | 1 |

Misses in library cache during parse: 1

Optimizer mode: ALL_ROWS

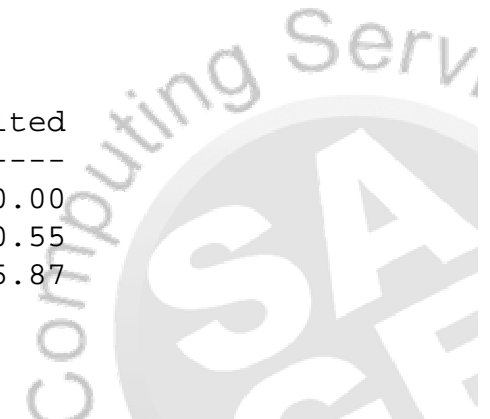
Parsing user id: 82

Rows Row Source Operation

```
-----
      1  SORT AGGREGATE (cr=35383 pr=35341 pw=35341 time=0 us)
398155  HASH JOIN (cr=35383 pr=35341 pw=35341 time=131437 us cost=9656 size=14112720
      card=392020)
19800   INDEX RANGE SCAN EVTLG_PK (cr=37 pr=0 pw=0 time=304 us cost=2 size=338470
      card=19910)(object id 69847)
398155  TABLE ACCESS FULL BOOKINGS_LARGE (cr=35346 pr=35341 pw=35341 time=107228 us
      cost=9652 size=7321631 card=385349)
```

Elapsed times include waiting on following events:

| Event waited on | Times Waited | Max. Wait | Total Waited |
|-----------------------------|-----------------|-----------|--------------|
| SQL*Net message to client | 2 | 0.00 | 0.00 |
| SQL*Net message from client | 2 | 30.55 | 30.55 |
| direct path read | 314 | 0.25 | 15.87 |



Automatic Tuning

However good this looks:

Let it do its automatic tuning

Don't let it automatically implement (in a transaction database)

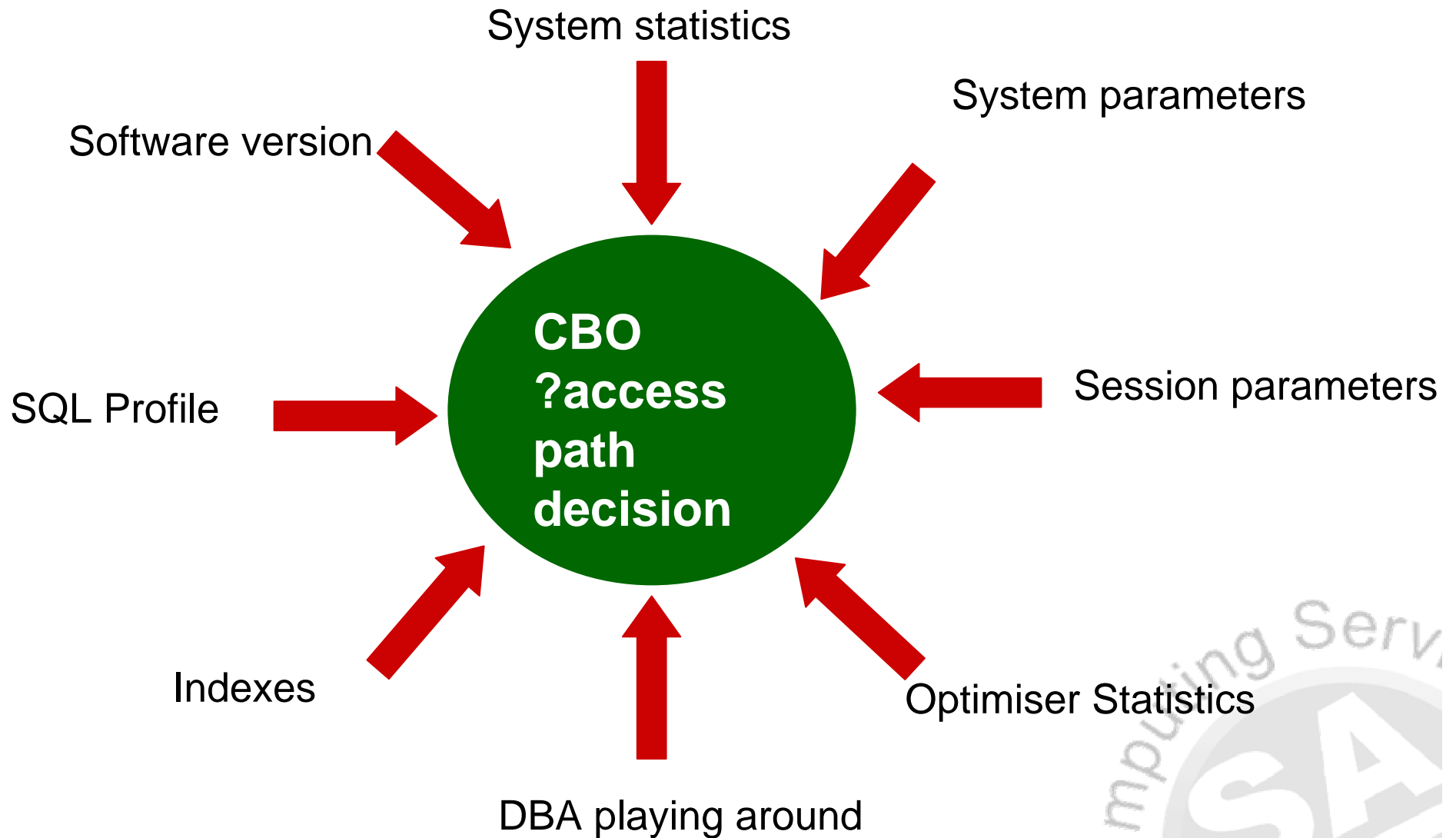
Check its recommendations and implement if you like them





SQL Plan Management

CBO - Instability



CBO - Instability

- Solution 1
 - Leave it all alone
- Solution 2
 - SQL Plan Management
 - Store plan baseline
 - Plans not used till accepted
 - Manually accept or
 - Allow Oracle to evolve plans

SQL Plan Management

Manual capture

DBMS_SPM.LOAD_PLANS_FROM_SQLSET

DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE

Auto capture of repeatable statements

OPTIMIZER_CAPTURE_SQL_PLAN_BASELINE = TRUE

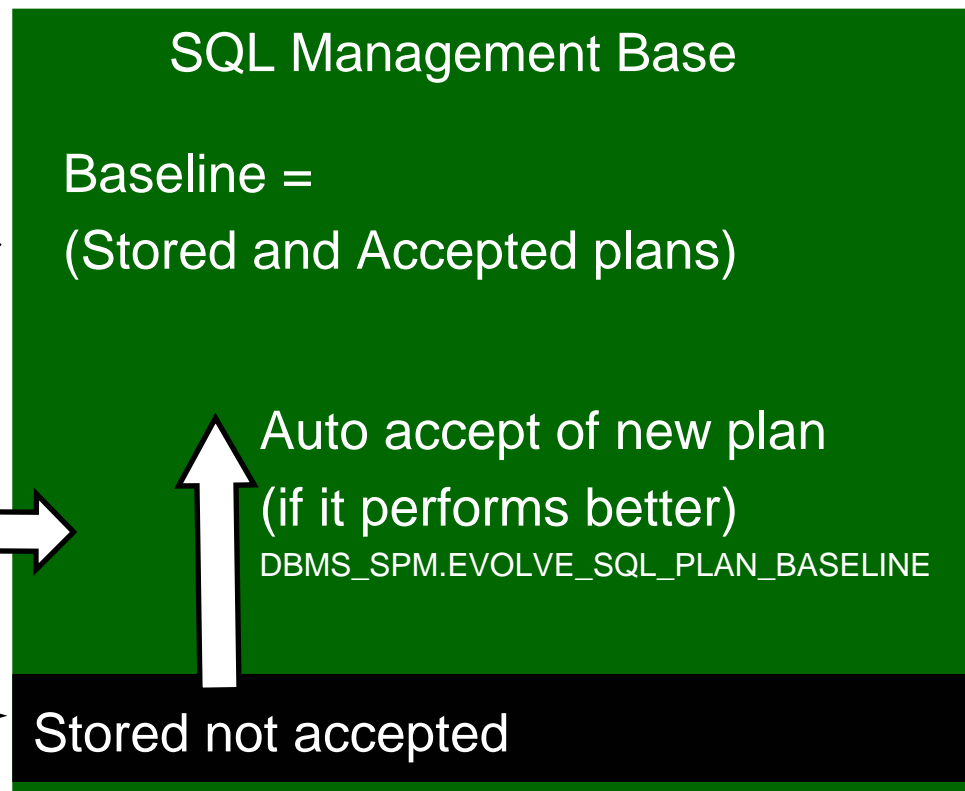
Manual load/accept of new plan

DBMS_SPM.LOAD_PLANS_FROM_SQLSET

DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE

SQL Tuning Advisor identifies new plan – SQL*Profile accepted

New Plan identified during execution



SQL Plan Management – Fixed Plans

Manual load/accept of new plan

DBMS_SPM.LOAD_PLANS_FROM_SQLSET

DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE

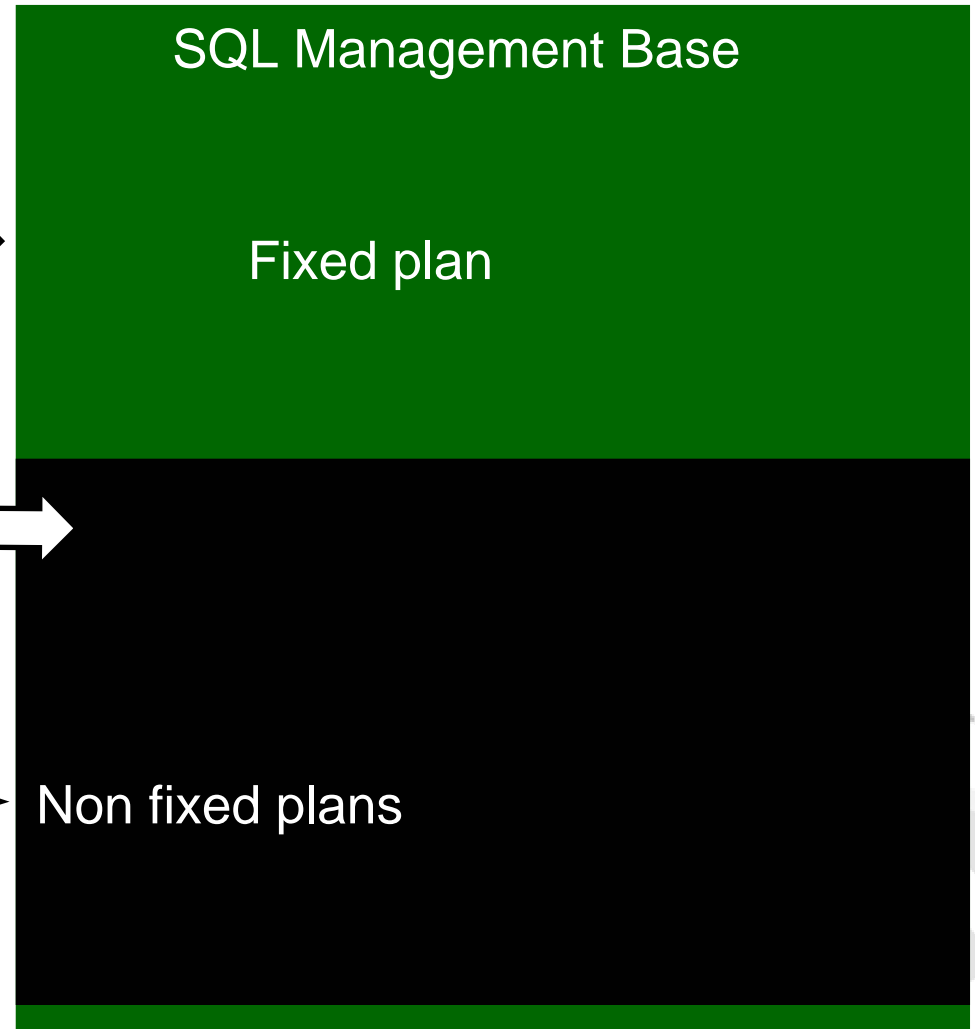
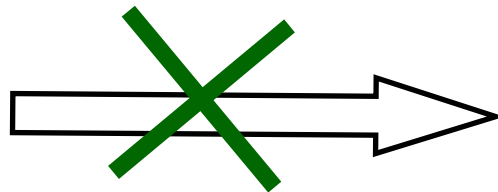
FIXED=> Y



SQL Tuning Advisor identifies new
plan – SQL*Profile accepted



New Plan
identified but
not added



SQL Plan Management

alter system flush shared_pool
No SQL Profiles, No baselines

Settings

Capture SQL Plan Baselines [TRUE](#)

Use SQL Plan Baselines [TRUE](#)

| | | |
|-----------------------|----|---------------------------|
| Plan Retention(Weeks) | 53 | Configure |
|-----------------------|----|---------------------------|

```
SELECT count(b.comments)
FROM events_large e, TRAIN.bookings_large b
WHERE e.event_no = b.event_no
AND e.event_no < 20000
AND b.resource code = 'BRLG'
```

Execute twice

alter system flush shared_pool

```
SELECT count(b.comments)
FROM events_large e, TRAIN.bookings_large b
WHERE e.event_no = b.event_no
AND e.event_no < 20000
AND b.resource code = 'BRLG'
```

```
SELECT sql_text, sql_id, child_number, hash_value,
       executions,sql_profile, sql_plan_baseline
FROM v$sql WHERE UPPER(sql_text) LIKE '%BOOKINGS%'
```

| SQL_TEXT | SQL_ID | CHILD_NUMBER | HASH_VALUE | EXECUTIONS | SQL_PROFILE | SQL_PLAN_BASELINE |
|--|---------------|--------------|------------|------------|------------------------|-------------------|
| 1 SELECT sql_text, sql_id, child_nu... | fr1c8g845srwt | 0 | 140271513 | 1 (null) | SYS_SQL_PLAN_848cb269k | |
| 2 SELECT count(b.comments) FRO... | 1pch0rhrzb5zf | 0 | 804624366 | 2 (null) | SYS_SQL_PLAN_d54cb9f75 | |

SQL handle: SYS_SQL_9dded8fad54cb9f7

SQL text: SELECT count(b.comments) FROM events_large e, TRAIN.bookings_large b
WHERE e.event_no = b.event_no AND e.event_no
b.resource_code = 'BRLG'

Plan name: SYS_SQL_PLAN_d54cb9f757e3b7a2

Enabled: YES Fixed: NO Accepted: YES Origin: AUTO-CAPTURE

Plan hash value: 2958936254

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |
|-----|-----------------------------|----------------|------|-------|-------------|----------|
| 0 | SELECT STATEMENT | | 1 | 24 | 16 (0) | 00:00:01 |
| 1 | SORT AGGREGATE | | 1 | 24 | | |
| 2 | NESTED LOOPS | | | | | |
| 3 | NESTED LOOPS | | 4 | 96 | 16 (0) | 00:00:01 |
| * 4 | INDEX RANGE SCAN | EVTLG_PK | 1 | 5 | 2 (0) | 00:00:01 |
| * 5 | INDEX RANGE SCAN | BK_EVT2 | 11 | | 2 (0) | 00:00:01 |
| * 6 | TABLE ACCESS BY INDEX ROWID | BOOKINGS_LARGE | 4 | 76 | 14 (0) | 00:00:01 |

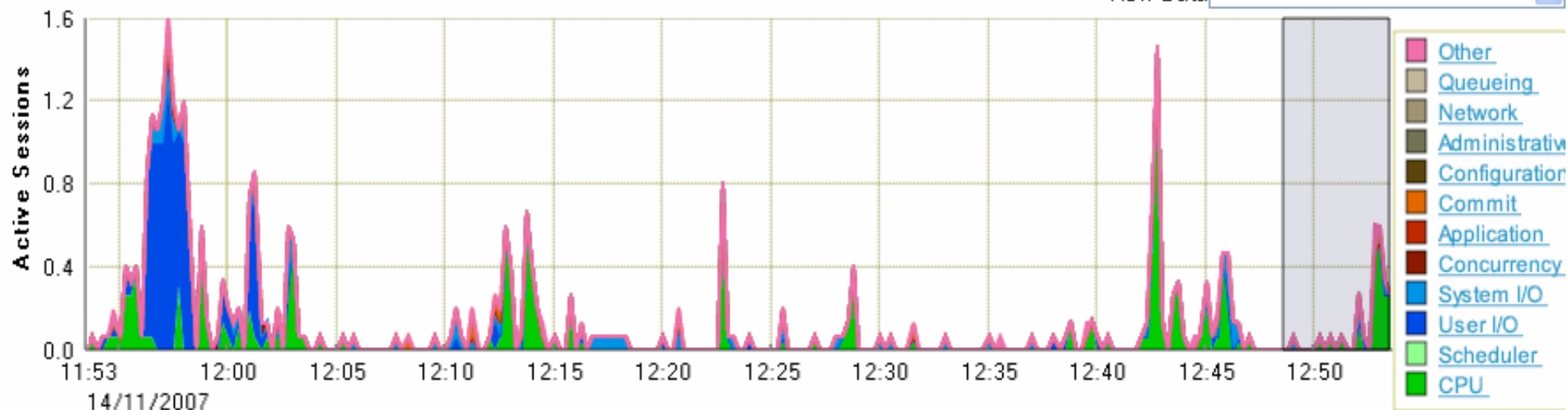
Predicate Information (identified by operation id):

- 4 - access("E"."EVENT_NO"
- 5 - access("E"."EVENT_NO"="B"."EVENT_NO")
 filter("B"."EVENT_NO"
- 6 - filter("B"."RESOURCE_CODE"='BRLG')

Top Activity

Drag the shaded box to change the time period for the detail section below.

View Data Real Time: 15 Second Refresh













Detail for Selected 5 Minute Interval

Start Time 14/11/2007 12:48:33 PM WST

Run ASH Report

Top SQL

| Actions | Schedule SQL Tuning Advisor |  |  |
|--|--|---|---|
| Select All Select None | | | |
| Select | Activity (%)  | SQL ID | SQL Type |
| <input checked="" type="checkbox"/> |  25.00 | 1pch0rhrzb5zf | SELECT |
| <input type="checkbox"/> |  17.86 | gxxa073u093s4 | SELECT |
| <input type="checkbox"/> |  7.14 | 26rx7huzvtuy0 | SELECT |
| <input type="checkbox"/> |  7.14 | 40u9awsftkwn4 | SELECT |
| <input type="checkbox"/> |  7.14 | bvf3fxv3hatw7 | SELECT |
| <input type="checkbox"/> |  7.14 | bvf3fxv3hatw7 | SELECT |
| <input type="checkbox"/> |  3.57 | 0pyavv0cxwyyv | SELECT |

Top Sessions

| View | Top Sessions | | | |
|-------------------|---------------------|------------------------|-------------------|---------|
| Activity (%) ▾ | | Session ID | User Name | Program |
| <div></div> 41.67 | 122 | DBSNMP | emagent.exe | |
| <div></div> 19.44 | 139 | TRAIN | SQL Developer | |
| <div></div> 11.11 | 158 | SYS | ORACLE.EXE (CKPT) | |
| <div></div> 5.56 | 114 | SYS | OMS | |
| <div></div> 5.56 | 148 | SYSMAN | ORACLE.EXE (J000) | |
| <div></div> 2.78 | 156 | SYS | ORACLE.EXE (MMON) | |
| <div></div> 2.78 | 138 | SYSMAN | OMS | |

Database Instance: ora11.sagecomputing.com.au > Advisor Central >
SQL Tuning Results:SQL_TUNING_1195016151984 >

Logged in As SYS

Recommendations for SQL ID:1pch0hrzb5zf

[Return](#)

Only one recommendation should be implemented.

SQL Text

[SELECT count\(b.comments\) FROM events_large e, TRAIN.bookings_large b WHERE e.event_no = b.event_no AND e.event_no < 20000 AND b.resource_code = 'BRLG'](#)

Select Recommendation

[Original Explain Plan \(Annotated\)](#)

[Implement](#)

| Select | Type | Findings | Recommendations | Rationale | Benefit (%) | New Explain Plan | Compare Explain Plans |
|--------|-------------|---|--|-----------|-------------|------------------|-----------------------|
| | SQL Profile | A potentially better execution plan was found for this statement. | Consider accepting the recommended SQL profile. A SQL plan baseline corresponding to the plan with the SQL profile will also be created. | | 96.26 | | |

[Return](#)

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

[About Oracle Enterprise Manager](#)

A SQL Plan Baseline is an execution plan deemed to have acceptable performance for a given SQL statement.

Settings

Capture SQL Plan Baselines [TRUE](#)

Use SQL Plan Baselines [TRUE](#)

Plan Retention(Weeks) [Configure](#)

Jobs for SQL Plan Baselines

[Pending](#)[Completed](#)[Load Jobs](#)

Search

SQL Text [Go](#)

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

[Load](#) [Unpack](#)

[Enable](#) [Disable](#) [Drop](#) [Evolve](#) [Pack](#) [Fixed - Yes](#) [Go](#)

[Select All](#) | [Select None](#)

| Select | Name ▾ | SQL Text | Enabled | Accepted | Fixed | Auto Purge | Created | Last Modified |
|--------------------------|---|---|---------|----------|-------|------------|------------------------|------------------------|
| <input type="checkbox"/> | SYS_SQL_PLAN_d54cb9f757e3b7a2 | SELECT count(b.comments) FROM events_large e, TRA... | YES | YES | NO | YES | 14/11/2007 12:42:42 | 14/11/2007 12:42:42 |
| <input type="checkbox"/> | SYS_SQL_PLAN_d54cb9f74045a280 | SELECT count(b.comments) FROM events_large e, TRA... | YES | YES | NO | YES | 14/11/2007 12:58:36 | 14/11/2007 12:58:36 |
| <input type="checkbox"/> | SYS_SQL_PLAN_848cb269b73cade2 | SELECT sql_text, sql_id, child_number, hash_value,... | YES | YES | NO | YES | 14/11/2007 12:43:55 | 14/11/2007 12:43:55 |

✓ **TIP** The table will display maximum of 2000 rows. Use search criteria to get the desired results.

SQL handle: SYS_SQL_9dded8fad54cb9f7

SQL text: SELECT count(b.comments) FROM events_large e, TRAIN.bookings_large b
WHERE e.event_no = b.event_no AND e.event_no
b.resource_code = 'BRLG'

Plan name: SYS_SQL_PLAN_d54cb9f74045a280

Enabled: YES Fixed: NO Accepted: YES Origin: MANUAL-SQLTUNE

Plan hash value: 2611015962

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |

| 0 | SELECT STATEMENT | | 1 | 24 | 9724 (1) | 00:01:57 |
| 1 | SORT AGGREGATE | | 1 | 24 | | |
|* 2 | HASH JOIN | | 391K | 9179K | 9724 (1) | 00:01:57 |
|* 3 | INDEX RANGE SCAN | EVTLG_PK | 19814 | 99070 | 2 (0) | 00:00:01 |
|* 4 | TABLE ACCESS FULL | BOOKINGS_LARGE | 408K | 7584K | 9720 (1) | 00:01:57 |

Predicate Information (identified by operation id):

- 2 - access("E"."EVENT_NO"="B"."EVENT_NO")
- 3 - access("E"."EVENT_NO"
- 4 - filter("B"."EVENT_NO"

Note

- SQL profile "SYS_SQLPROF_01458f2179798002" used for this statement

SQL Plan Management

```
SELECT count(b.comments)
FROM events_large e, TRAIN.bookings_large b
WHERE e.event_no = b.event_no
AND e.event_no < 20000
AND b.resource_code = 'BRLG'
```

```
SELECT sql_text, sql_id, child_number, hash_value,
       executions,sql_profile, sql_plan_baseline
FROM v$sql WHERE UPPER(sql_text) LIKE '%BOOKINGS%'
```

| Results | | | | | | | |
|--|--------------------------------------|---------------|--------------|------------|-------------------|-------------|-------------------------------|
| Script Output Explain Autotrace DBMS Output OWA Output | | | | | | | |
| Results: | | | | | | | |
| | SQL_TEXT | SQL_ID | CHILD_NUMBER | HASH_VALUE | EXECUTIONS | SQL_PROFILE | SQL_PLAN_BASELINE |
| 1 | SELECT sql_text, sql_id, child_nu... | fr1c8g845srwt | 0 | 140271513 | 2 (null) | | SYS_SQL_PLAN_848cb269b73cade2 |
| 2 | SELECT count(b.comments) FRO... | 1pch0hrzb5zf | 0 | 804624366 | 1 SYS_SQLPROF_... | | SYS_SQL_PLAN_d54cb9f74045a280 |

SQL Plan Management – Plan Retention

- Default 53 weeks
- Only unused plans purged
- Why would you keep a plan for a year if it has only been used once

```
BEGIN DBMS_SPM.CONFIGURE(  
  'plan_retention_weeks',13);  
END;
```



SQL Plan Management – Binds

- Will not automatically handle adaptive cursors
- New plan identified on first execution
- New plan recorded as not accepted
- Plan will not evolve
- All bind variable values use same baseline plan
- Plans show as not bind sensitive or aware

| SQL_TEXT | SQL_ID | CHILD_... | HASH_VALUE | EXECUTIONS | SQ... | SQL_PLAN_BASELINE |
|---------------------------|---------------|-----------|------------|------------|--------|-------------------------------|
| SELECT COUNT(quantity)... | 2wpuu88g5an3m | 0 | 508907635 | 2 (null) | (null) | |
| SELECT COUNT(quantity)... | 2wpuu88g5an3m | 1 | 508907635 | 2 (null) | | SYS_SQL_PLAN_845e4b385166fb0e |
| SELECT COUNT(quantity)... | 2wpuu88g5an3m | 2 | 508907635 | 2 (null) | | SYS_SQL_PLAN_845e4b385166fb0e |

| SQL_ID | CHILD_NUMBER | IS_BIND_SENSITIVE | IS_BIND_AWARE |
|---------------|--------------|-------------------|---------------|
| 2wpuu88g5an3m | 0 | N | N |
| 2wpuu88g5an3m | 1 | N | N |
| 2wpuu88g5an3m | 2 | N | N |

Plan name: SYS_SQL_PLAN_845e4b38128d5d7c
Enabled: YES Fixed: NO Accepted: NO Origin: AUTO-CAPTURE

Plan hash value: 927983165

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |

| 0 | SELECT STATEMENT | | 1 | 8 | 9717 (1) | 00:01:57 |
| 1 | SORT AGGREGATE | | 1 | 8 | | |
| * 2 | TABLE ACCESS FULL | BOOKINGS_LARGE | 640K | 5006K | 9717 (1) | 00:01:57 |

Predicate Information (identified by operation id):

2 - filter("RESOURCE_CODE"=:V3)

Plan name: SYS_SQL_PLAN_845e4b385166fb0e
Enabled: YES Fixed: NO Accepted: YES Origin: AUTO-CAPTURE

Plan hash value: 1457437069

| Id | Operation | Name | Rows | Bytes | Cost (%CPU) | Time |

| 0 | SELECT STATEMENT | | 1 | 8 | 35565 (1) | 00:07:07 |
| 1 | SORT AGGREGATE | | 1 | 8 | | |
| 2 | TABLE ACCESS BY INDEX ROWID | BOOKINGS_LARGE | 640K | 5006K | 35565 (1) | 00:07:07 |
| * 3 | INDEX RANGE SCAN | BK_RES2 | 640K | | 1764 (1) | 00:00:22 |

Evolve SQL Plan Baseline Report

Inputs:

PLAN_LIST = SYS_SQL_PLAN_845e4b38128d5d7c
TIME_LIMIT = DBMS_SPM.AUTO_LIMIT
VERIFY = YES
COMMIT = YES

Plan: SYS_SQL_PLAN_845e4b38128d5d7c

Plan was verified: Time used 107.25 seconds.

Failed performance criterion: Compound improvement ratio <= 1.2.

| | Baseline Plan | Test Plan | Improv. Ratio |
|-------------------|---------------|-----------|---------------|
| | ----- | ----- | ----- |
| Execution Status: | COMPLETE | COMPLETE | |
| Rows Processed: | 1 | 1 | |
| Elapsed Time(ms): | 51902 | 10310 | 5.03 |
| CPU Time(ms): | 4078 | 1281 | 3.18 |
| Buffer Gets: | 42327 | 35384 | 1.2 |
| Disk Reads: | 23204 | 14800 | 1.57 |
| Direct Writes: | 0 | 0 | |
| Fetches: | 23201 | 1018 | 22.79 |
| Executions: | 1 | 1 | |

Report Summary

Number of SQL plan baselines verified: 1.

Number of SQL plan baselines evolved: 0.

Database Instance: ora11.sagecomputing.com.au >

SQL Plan Control

Evolve SQL Plan Baselines

Plans that have not yet been accepted can be evolved (verified) to confirm they are suitable plan baselines.

| Name | SQL Text |
|-------------------------------|---|
| SYS_SQL_PLAN_845e4b38128d5d7c | SELECT COUNT(quantity) FROM bookings_large WHERE... |

Verify Performance ☐ Yes ☒ No

Time Limit ☒ Auto ☐ Limited ☐ Specify (minutes)

Action ☒ Report and Accept ☐ Report only

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.





[About Oracle Enterprise Manager](#)

Force Acceptance of the plan

SQL Plan Management – Binds

Once you have forced the plan to be accepted the statement will be bind aware

| | | | |
|---|---|-----|-----|
| SYS SQL PLAN 845e4b385166fb0e | SELECT COUNT(quantity) FROM bookings large WHERE... | YES | YES |
| SYS SQL PLAN 845e4b38128d5d7c | SELECT COUNT(quantity) FROM bookings large WHERE... | YES | YES |

|  SQL_ID |  CHILD_NUMBER |  IS_BIND_SENSITIVE |  IS_BIND_AWARE |
|--|--|---|---|
| 2wpuu88g5an3m | 0 | Y | N |
| 2wpuu88g5an3m | 1 | Y | Y |

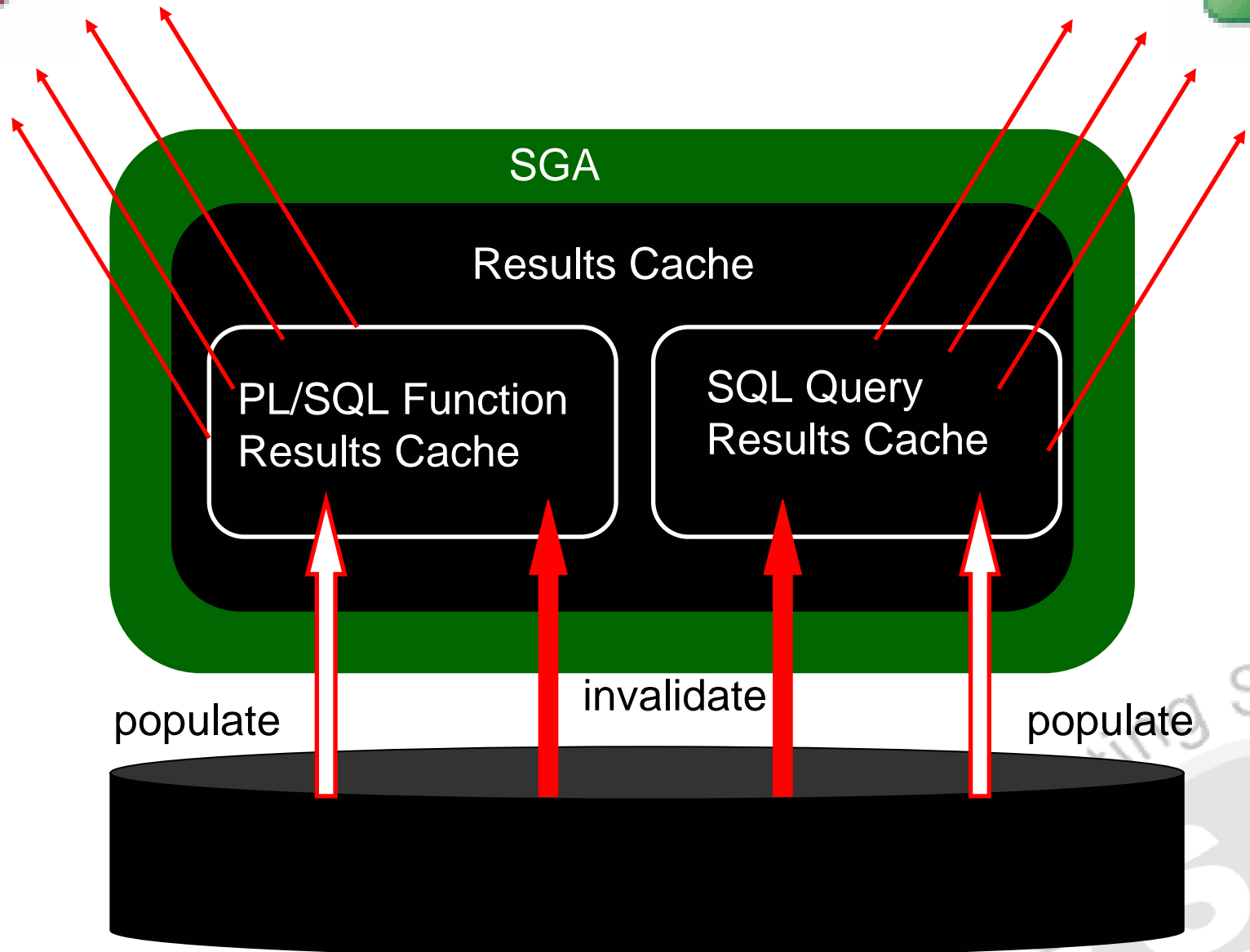
|  SQL_TEXT |  SQL_ID |  CHILD_... |  HASH_VALUE |  EXECUTIONS |   SQL_PLAN_BASELINE |
|--|--|---|--|--|---|
| SELECT COUNT(quantity)... | 2wpuu88g5an3m | 0 | 508907635 | 4 ... | SYS_SQL_PLAN_845e4b385166fb0e |
| SELECT COUNT(quantity)... | 2wpuu88g5an3m | 1 | 508907635 | 2 ... | SYS_SQL_PLAN_845e4b38128d5d7c |



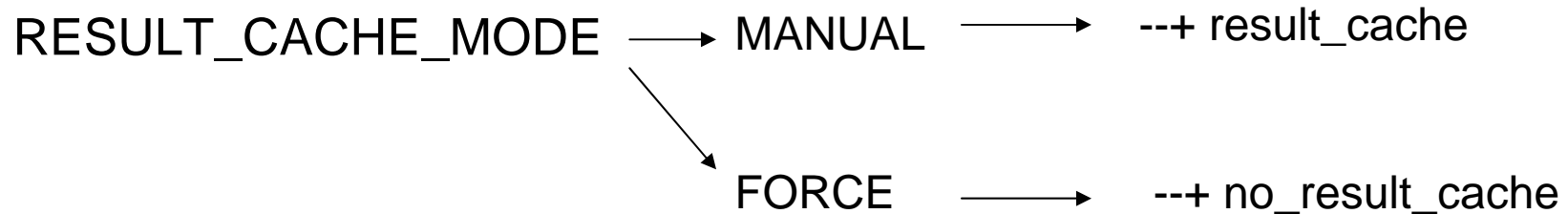


Cached Query Results

Results Cache



Query Results Cache



Will not cache if:-

- Dictionary tables

- Temporary tables




- Sequences

- Dates/times

- Non deterministic PL/SQL

Monitoring the Results Cache

- SELECT * FROM v\$result_cache_memory
- SELECT * FROM v\$result_cache_objects
- SELECT * FROM v\$result_cache_statistics
- SELECT * FROM v\$result_cache_dependency
- RESULT_CACHE_MAX_SIZE
- RESULT_CACHE_MAX_RESULT
- Explain Plan

|  OPERATION |  OPTIONS |  OBJECT_NAME |
|---|---|---|
| SELECT STATEMENT | (null) | (null) |
| RESULT CACHE | (null) | bzurnr2sm456r4yctszyqks1s5 |
| SORT | AGGREGATE | (null) |
| NESTED LOOPS | (null) | (null) |
| TABLE ACCESS | FULL | BOOKINGS_LARGE |
| INDEX | UNIQUE SCAN | EVTLG_PK |



Query Results Cache

SELECT * FROM v\$result_cache_objects

| ID | TYPE | STATUS | BUCKET_NO | HASH | NAME | NAM |
|-------|--------|-----------|-----------|------------|--|-----|
| 8206 | Result | Published | 2159 | 1148708975 | SELECT CMD.COMMAND_NAME, CMD.COMM... | SQL |
| 8204 | Result | Published | 3630 | 1492413998 | SELECT COUNT(1) FROM MGMT_JOB_PARA... | SQL |
| 8203 | Result | Published | 2136 | 2272741464 | SELECT STEP_NAME, STEP_TYPE FROM MGM... | SQL |
| 8202 | Result | Published | 2221 | 3687729325 | SELECT COUNT(1) FROM MGMT_JOB_PARA... | SQL |
| 8201 | Result | Published | 3193 | 2885282937 | SELECT STEP_NAME, STEP_TYPE FROM MGM... | SQL |
| 81933 | Result | Published | 3671 | 2526297687 | SELECT COUNT(1) FROM MGMT_JOB_BLACK... | SQL |
| 8195 | Result | Published | 519 | 286777863 | SELECT PRIVILEGE_NAME, PRIVILEGE_TYPE, ... | SQL |
| 8194 | Result | Published | 2147 | 4272883811 | SELECT SUSPEND_ON_NOCREDS FROM MGM... | SQL |
| 8193 | Result | Published | 3736 | 1075990168 | SELECT JOB_TYPE_CATEGORY FROM MGMT... | SQL |
| 8192 | Result | Published | 2146 | 1426720866 | SELECT MGMT_JOB_PARAMSRC_RECORD(S... | SQL |
| 86022 | Result | Published | 2338 | 2963659042 | SELECT /*+ ORDERED PUSH_SUBQ */ DISTINC... | SQL |
| 4119 | Result | Published | 3118 | 3553827886 | SELECT SINGLE_TARGET FROM MGMT_JOB_... | SQL |
| 69644 | Result | Published | 3657 | 1467706953 | SELECT /*+ ORDERED PUSH_SUBQ */ DISTINC... | SQL |
| 4108 | Result | Published | 2358 | 514570550 | SELECT JOB_ID FROM MGMT_JOB_NOTIFY_S... | SQL |

PL/SQL Function Results Cache

```
CREATE OR REPLACE FUNCTION quantity_booked
(p_resource_code in resources.code%TYPE,p_event_date in date)
RETURN NUMBER
RESULT_CACHE RELIES_ON (BOOKINGS,EVENTS)
IS
    v_total_booked number := 0;
BEGIN
    SELECT      sum(b.quantity)
    INTO        v_total_booked
    FROM        bookings b, events e
    WHERE       e.event_no = b.event_no
    AND         p_event_date between e.start_date and e.end_date
    AND         b.resource_code = p_resource_code;

    book_pkg.add_one(v_total_booked, v_total_booked);

    RETURN (v_total_booked);
END;
```



PL/SQL Function Results Cache

```
DECLARE
```

```
v_num number;
```

```
BEGIN
```

```
  v_num := quantitybooked('VCR2','06-NOV-2007');
```

```
  dbms_output.put_line(v_num);
```

```
END;
```

9

```
SELECT *
```

```
FROM v$result_cache_objects
```

| NAME | TYPE | STATUS | NAME | PIN_CODE | SCAN_COUNT | INVALIDATIONS |
|--------|------------|-----------|---|----------|------------|---------------|
| NAME | TYPE | STATUS | NAME | PIN_CODE | SCAN_COUNT | INVALIDATIONS |
| (null) | Dependency | Published | TRAIN.QUANTITYBOOKED | 0 | 0 | 0 |
| (null) | Dependency | Published | TRAIN.BOOKINGS | 0 | 0 | 0 |
| (null) | Dependency | Published | TRAIN.EVENTS | 0 | 0 | 0 |
| PLSQL | Result | Published | "TRAIN"."QUANTITYBOOKED":8."QUANTITYBOOKED"#496944ef298963c5 #1 | 0 | 1 | 0 |

PL/SQL Function Results Cache

```
UPDATE bookings SET quantity = quantity*2;
COMMIT;
```

| N... | TYPE | STATUS | NAME | PIN_CO... | SCAN_COUNT | INVALIDATIONS |
|--------|------------|-----------|--|-----------|------------|---------------|
| (null) | Dependency | Published | TRAIN.QUANTITYBOOKED | 0 | 0 | 0 |
| (null) | Dependency | Published | TRAIN.BOOKINGS | 0 | 0 | 1 |
| (null) | Dependency | Published | TRAIN.EVENTS | 0 | 0 | 0 |
| PLSQL | Result | Invalid | "TRAIN"."QUANTITYBOOKED"::8."QUANTITYBOOKED"#496944ef298963c5 #1 | 0 | 1 | 0 |

```
DECLARE
```

```
v_num number;
```

```
BEGIN
```

```
  v_num := quantitybooked('VCR2','06-NOV-2007');
```

```
  dbms_output.put_line(v_num);
```

```
END;
```

17

| N... | TYPE | STATUS | NAME | PIN_CO... | SCAN_COUNT | INVALIDATIONS |
|--------|------------|-----------|--|-----------|------------|---------------|
| (null) | Dependency | Published | TRAIN.QUANTITYBOOKED | 0 | 0 | 0 |
| (null) | Dependency | Published | TRAIN.BOOKINGS | 0 | 0 | 1 |
| (null) | Dependency | Published | TRAIN.EVENTS | 0 | 0 | 0 |
| PLSQL | Result | Published | "TRAIN"."QUANTITYBOOKED"::8."QUANTITYBOOKED"#496944ef298963c5 #1 | 0 | 0 | 0 |
| PLSQL | Result | Invalid | "TRAIN"."QUANTITYBOOKED"::8."QUANTITYBOOKED"#496944ef298963c5 #1 | 0 | 1 | 0 |

PL/SQL Function Results Cache

```
CREATE OR REPLACE PACKAGE BODY
  book_pkg AS
PROCEDURE add_one (p_result IN NUMBER,
  p_result_out OUT NUMBER)
IS
BEGIN
  p_result_out := p_result+10;
END;
END;
```

| N... | TYPE | STATUS | NAME | PIN_CO... | SCAN_COUNT | INVALIDATIONS |
|--------|------------|-----------|--|-----------|------------|---------------|
| (null) | Dependency | Published | TRAIN.QUANTITYBOOKED | 0 | 0 | 0 |
| (null) | Dependency | Published | TRAIN.BOOKINGS | 0 | 0 | 1 |
| (null) | Dependency | Published | TRAIN.EVENTS | 0 | 0 | 0 |
| PLSQL | Result | Published | "TRAIN"."QUANTITYBOOKED"::8."QUANTITYBOOKED"#496944ef298963c5 #1 | 0 | 1 | 0 |
| PLSQL | Result | Invalid | "TRAIN"."QUANTITYBOOKED"::8."QUANTITYBOOKED"#496944ef298963c5 #1 | 0 | 1 | 0 |

PL/SQL Function Results Cache

DECLARE

v_num number;

BEGIN

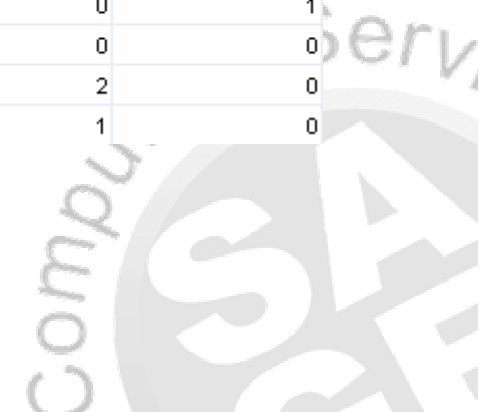
v_num := quantitybooked('VCR2','06-NOV-2007');

dbms_output.put_line(v_num);

END;

17

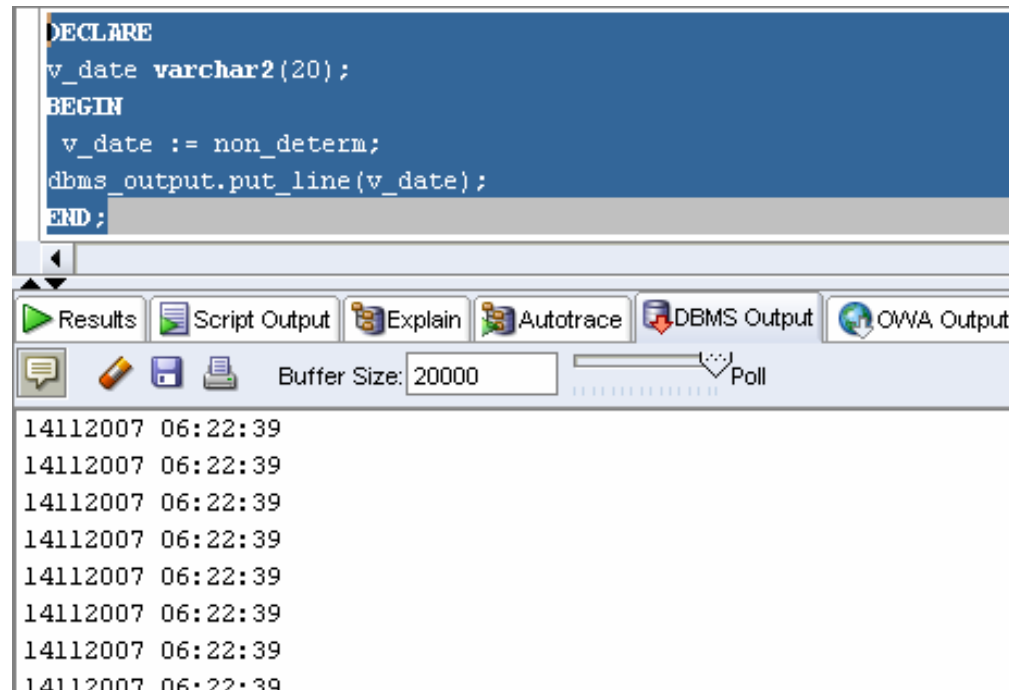
| N... | TYPE | STATUS | NAME | PIN_CO... | SCAN_COUNT | INVALIDATIONS |
|--------|------------|-----------|--|-----------|------------|---------------|
| {null} | Dependency | Published | TRAIN.QUANTITYBOOKED | 0 | 0 | 0 |
| {null} | Dependency | Published | TRAIN.BOOKINGS | 0 | 0 | 1 |
| {null} | Dependency | Published | TRAIN.EVENTS | 0 | 0 | 0 |
| PLSQL | Result | Published | "TRAIN"."QUANTITYBOOKED"::8."QUANTITYBOOKED"#496944ef298963c5 #1 | 0 | 2 | 0 |
| PLSQL | Result | Invalid | "TRAIN"."QUANTITYBOOKED"::8."QUANTITYBOOKED"#496944ef298963c5 #1 | 0 | 1 | 0 |



PL/SQL Function Results Cache

Don't do stupid stuff

```
CREATE OR REPLACE FUNCTION non_determ
RETURN VARCHAR2 RESULT_CACHE
IS
BEGIN
    RETURN
    TO_CHAR(sysdate,'ddmmyyyy hh:mi:ss');
END;
```



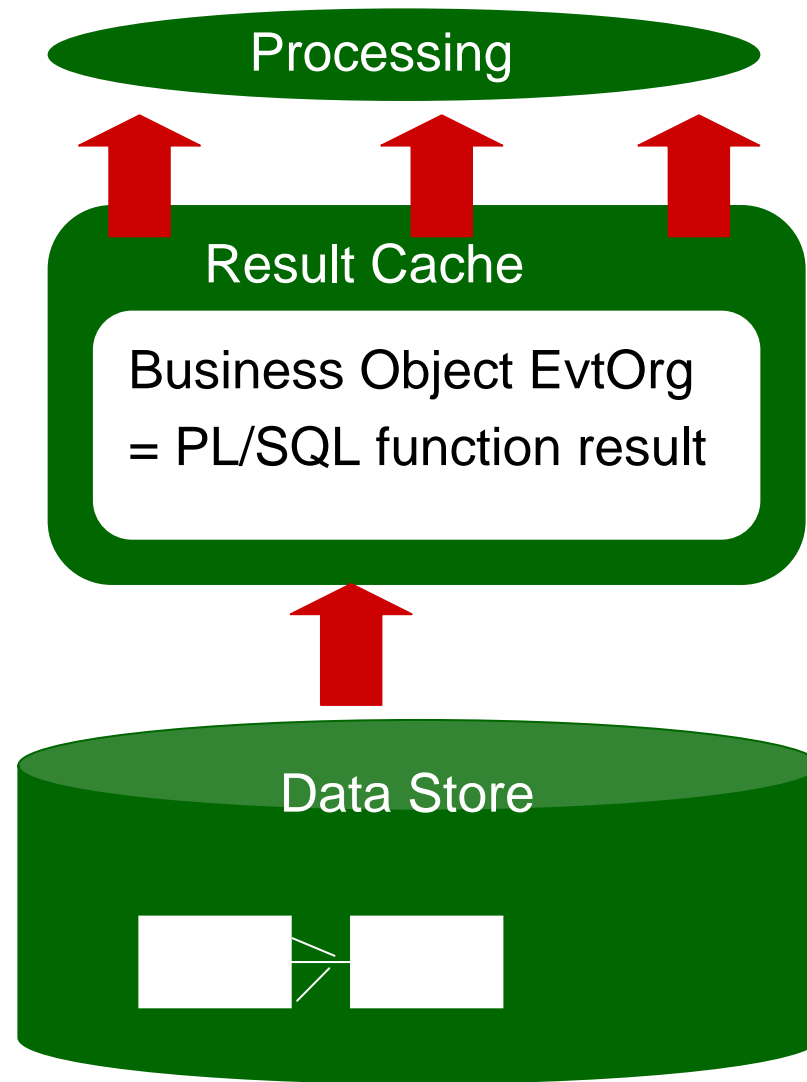
```
DECLARE
v_date varchar2(20);
BEGIN
    v_date := non_determ;
    dbms_output.put_line(v_date);
END;
```

Results | Script Output | Explain | Autotrace | DBMS Output | OWA Output

Buffer Size: 20000 | Poll

```
14112007 06:22:39
14112007 06:22:39
14112007 06:22:39
14112007 06:22:39
14112007 06:22:39
14112007 06:22:39
14112007 06:22:39
14112007 06:22:39
```

Designing for the PL/SQL Function Results Cache



Designing for the PL/SQL Function Results Cache

```
CREATE OR REPLACE TYPE VieworgEvtRec IS OBJECT (org_id number(4),
        name          varchar2(35),
        state          varchar2(3),
        event_no        number,
        start_date      date,
        end_date         date,
        contact_name    varchar2(30));
```

```
CREATE OR REPLACE TYPE VieworgEvtTab IS TABLE OF VieworgEvtRec;
```

```
PACKAGE BusinessObjects AS
```

```
    TYPE orgEvtRec IS RECORD (org_id organisations.org_id%TYPE,
        name          organisations.name%TYPE,
        state          organisations.state%TYPE,
        event_no        events.event_no%TYPE,
        start_date      events.start_date%TYPE,
        end_date         events.end_date%TYPE,
        contact_name    events.contact_name%TYPE);
```

```
    TYPE orgEvtTab IS TABLE OF orgEvtRec;
```

```
Function OrgEvents (p_org_id in organisations.org_id%TYPE)
RETURN orgEvtTab
RESULT_CACHE;
```

```
Function getOrgEvents (p_org_id in organisations.org_id%TYPE)
RETURN VieworgEvtTab;
```

```
END BusinessObjects;
```



Designing for the PL/SQL Function Results Cache

```
CREATE OR REPLACE PACKAGE BODY BusinessObjects AS
Function OrgEvents (p_org_id in organisations.org_id%TYPE)
RETURN orgEvtTab RESULT_CACHE RELIES_ON (ORGANISATIONS, EVENTS) IS
    CURSOR c_org IS
        SELECT o.org_id, o.name, o.state, e.event_no, e.start_date, e.end_date, e.contact_name
        FROM   organisations o, events e
        WHERE  o.org_id = e.org_id
        AND    o.org_id = p_org_id;
    t_orgEvt orgEvtTab;

BEGIN
    OPEN c_org;
    FETCH c_org BULK COLLECT INTO t_orgEvt;
    CLOSE c_org;
    RETURN t_orgEvt;
END OrgEvents;

Function getOrgEvents (p_org_id in organisations.org_id%TYPE)
RETURN VieworgEvtTab IS
    t_evt      VieworgEvtTab;
    t_evt_src  orgEvtTab;
BEGIN
    t_evt_src := OrgEvents(p_org_id);
    t_evt := VieworgEvtTab();
    FOR n in 1..t_evt_src.count LOOP
        t_evt.extend;
        t_evt(n) := VieworgEvtRec(t_evt_src(n).org_id, t_evt_src(n).name, t_evt_src(n).state,
            t_evt_src(n).event_no, t_evt_src(n).start_date, t_evt_src(n).end_date, t_evt_src(n).contact_name)
    END LOOP;
    RETURN t_evt;
END getOrgEvents;
END BusinessObjects;
```

Designing for the PL/SQL Function Results Cache

```
SELECT *
```

```
FROM TABLE (businessobjects.getOrgEvents(1000))
```

| TYPE | STATUS | NAME | SCAN_COUNT |
|------------|-----------|--|------------|
| Dependency | Published | TRAIN.BUSINESSOBJECTS | 0 |
| Dependency | Published | TRAIN.EVENTS | 0 |
| Dependency | Published | TRAIN.ORGANISATIONS | 0 |
| Result | Published | "TRAIN"."BUSINESSOBJECTS":11."ORGEVENTS"#46545659aaa98818 #4 | 9 |

So why do we need a middle tier?



SQL Performance Analyzer

Database Instance: ora11.sagecomputing.com.au >

Logged in As SYS

Confirmation

SQL tuning set TEST1 has been created successfully. A job CREATE_STS_1195021884609 to load SQL statements into the SQL tuning set TEST1 has been created successfully.

[View Job Details](#)

SQL Tuning Sets

Page Refreshed 14/11/2007 03:31:38 PM WST [Refresh](#)

A SQL Tuning Set is a collection of SQL Statements that can be used for tuning purposes.

Search [Go](#)
Filter on a name or partial name

[Create](#) [Import](#)

| Details Drop Export Schedule SQL Tuning Advisor Schedule SQL Access Advisor | | | | | | |
|---|-----------------------|--------|-------------|-----------|----------------|-----------------|
| Select | Name | Schema | Description | SQL Count | Created | Last Modified ▾ |
| <input checked="" type="radio"/> | TEST1 | SYS | | 0 | 14/11/07 14:31 | 14/11/07 14:31 |

Related Links

[SQL Performance Analyzer](#)

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

[About Oracle Enterprise Manager](#)



Internet

Database Instance: ora11.sagecomputing.com.au > [Advisor Central](#) > [SQL Performance Analyzer](#) >

Logged in As SYS

Guided Workflow

Page Refreshed 14/11/2007 15:33:20 [Refresh](#) [View Data](#) Real Time: 15 Second Refresh [▼](#)

The following guided workflow contains the sequence of steps necessary to execute a successful two-trial SQL Performance Analyzer test.

Note: Be sure that the Trial environment matches the tests you want to conduct.

| Step | Description | Executed | Status | Execute |
|------|--|----------|--------|---------|
| 1 | Create SQL Performance Analyzer Task based on SQL Tuning Set | | | |
| 2 | Replay SQL Tuning Set in Initial Environment | | | |
| 3 | Replay SQL Tuning Set in Changed Environment | | | |
| 4 | Compare Step 2 and Step 3 | | | |
| 5 | View Trial Comparison Report | | | |

TIP For an explanation of the icons and symbols used in the following table, see the [Icon Key](#)

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

[About Oracle Enterprise Manager](#)

Database Instance: ora11.sagecomputing.com.au > Advisor Central > SQL Performance Analyzer >

Guided Workflow: SYS.TEST1 >

Logged in As SYS

Create Replay Trial

Cancel

Replay Trials capture execution performance of the STS under a given optimizer environment.

SQL Performance Analyzer Task **SYS.TEST1**

SQL Tuning Set **SYS.TEST1**

* Replay Trial Name

Replay Trial Description

Per-SQL Time Limit

☒ **TIP** Time limit is on elapsed time of test execution of SQL.
EXPLAIN ONLY generates plans without test execution.

Trial environment determines results

The SQL Tuning Set remains constants under the SQL Performance Analyzer Task and its SQL is executed in isolation to create each Replay Trial. Performance differences between trials are thus attributed to environmental differences between trials.

Environmental changes affecting SQL optimization and performance may need to be made manually prior to execution of the Trial. These could include changing initialization parameters, gathering or setting optimizer statistics and creating indexes.

NOTE: Be sure trial environment has been established prior to submitting.


☐ Trial environment established

Schedule

Time Zone

☒ Immediately

☐ Later

Date 
(example: 14/11/2007)

Time ☐ AM ☒ PM

Cancel Submit

Database Instance: ora11.sagecomputing.com.au > Advisor Central > SQL Performance Analyzer >






Logged in As SYS

Guided Workflow: SYS.TEST1

Page Refreshed 14/11/2007 15:44:00 View Data Real Time: 15 Second Refresh

The following guided workflow contains the sequence of steps necessary to execute a successful two-trial SQL Performance Analyzer test.

Note: Be sure that the Trial environment matches the tests you want to conduct.

| Step | Description | Executed | Status | Execute |
|------|--|---------------------|--------|---|
| 1 | Create SQL Performance Analyzer Task based on SQL Tuning Set | 14/11/2007 14:34:08 | ✓ |  |
| 2 | Replay SQL Tuning Set in Initial Environment | 14/11/2007 14:37:29 | ✓ |  |
| 3 | Replay SQL Tuning Set in Changed Environment | 14/11/2007 14:41:49 | ✓ |  |
| 4 | Compare Step 2 and Step 3 | | ■ |  |
| 5 | View Trial Comparison Report | | ■ |  |

✓ **TIP** For an explanation of the icons and symbols used in the following table, see the [Icon Key](#)

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2007, Oracle. All rights reserved.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

[About Oracle Enterprise Manager](#)

Database Instance: ora11.sagecomputing.com.au > Advisor Central > SQL Performance Analyzer >
Guided Workflow: SYS.TEST1 >

Logged in As SY

Run Replay Trial Comparison

[Cancel](#) [Submit](#)

Task Name **SYS.TEST1**

SQL Tuning Set **SYS.TEST1**

Trial 1 Name SQL_REPLAY_1195022103937 ▼

Description

SQL Executed **Yes**

Trial 2 Name SQL_REPLAY_1195022398812 ▼

Description

SQL Executed **Yes**

Comparison Metric Elapsed Time ▼

Compare trials to assess change impact

SQL Performance Analyzer trial comparison allows you to assess the impact on SQL Tuning Set performance of changes made between two trials.

It is important to know the difference between Trial 1 and Trial 2 execution environments in order to properly assign impacts to the changes between trials. Tracking environmental changes between trials is currently a user responsibility.


The selected comparison metric is used as the basis for comparison, and defaults to execute elapsed time when both trials contain test execution statistics. When execution statistics are not available, a less accurate comparison can be made using optimizer cost.

Schedule

Time Zone Australia/Perth ▼

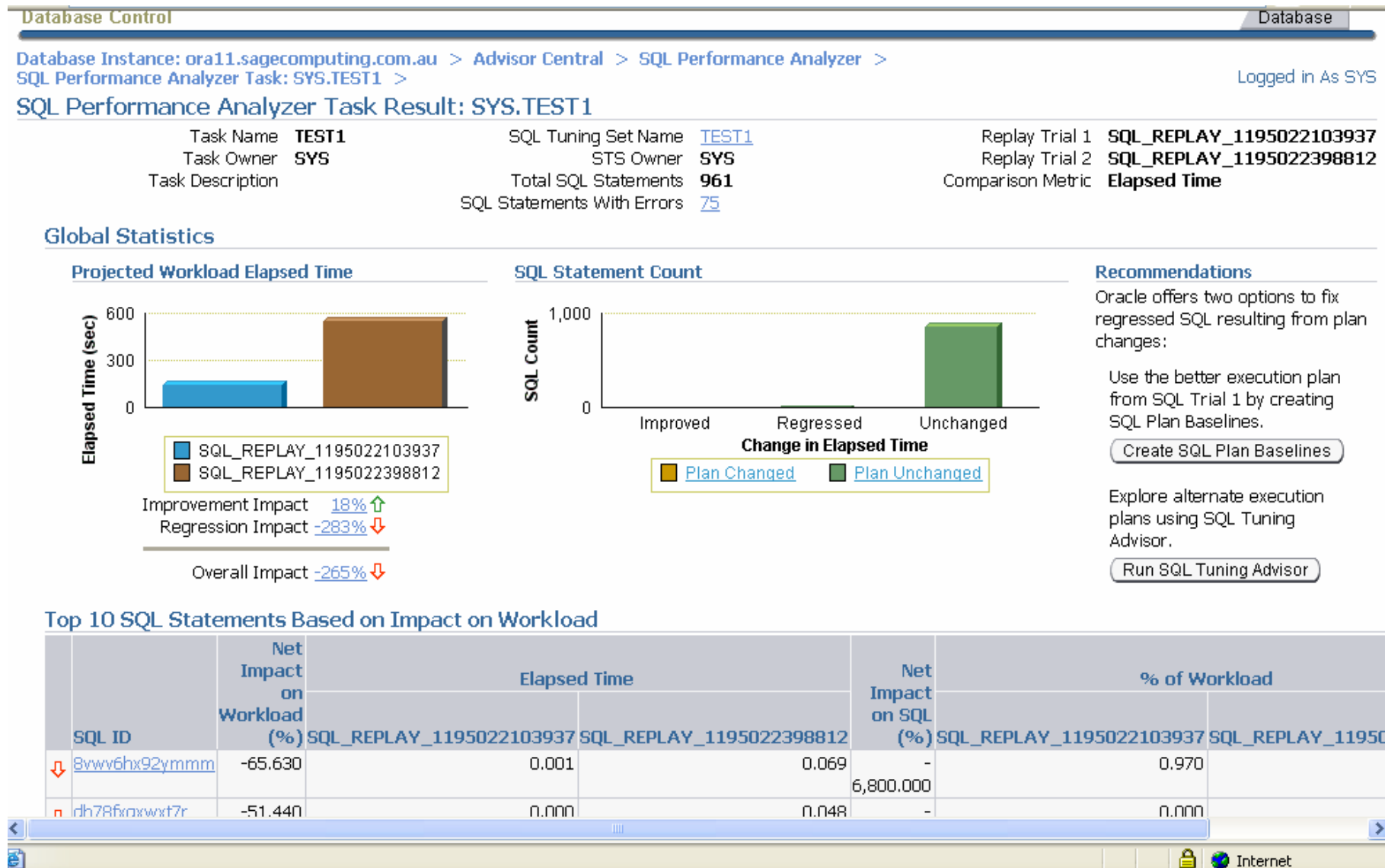
☒ Immediately

☐ Later

Date 14/11/2007 
(example: 14/11/2007)

Time 3 ▼ 33 ▼ 20 ▼ ☐ AM ☒ PM

View the Comparison Report



Summary - Setup

Migrate to Test instance of 11g

Run application

Collect SQL Tuning set

Run SQL Performance Analyzer

Add extra statistics

Run automatic tuning – without automatic implementation

Look at suggestions

fix the problem or

accept the profile

Lock any statistics where required (e.g very dynamic tables)

Run SQL Tuning set in SQL Performance Analyzer and compare

For a transaction database

Baseline the application SQL

If you are really paranoid fix the baseline plans



Summary

WORTH A TRIAL

Selectively cache PL/SQL results

If you have lots of spare memory turn results caching on



Summary – Maintenance (Transaction DB)

Keep auto stats gathering on (at a sensible interval), or manually gather stats when you have a quiet evening

Publish stats if performance not worse

Keep automatic tuning scheduled– automatic implementation off

Look at suggestions

fix the problem or

accept the profile

Monitor new baseline plans

manually test and accept

Before you change anything (your hairstyle/shirt/partner)

Run SQL Performance Analyzer and compare old and new configuration



Summary – Maintenance (Reporting DB)

Keep auto stats gathering on (at a sensible interval), or manually gather stats when you have a quiet evening

Publish stats if performance not worse

Keep automatic tuning scheduled – automatic implementation on

Before you make significant changes (version)

Run SQL Performance Analyzer and compare old and new configuration





SAGE Computing Services

Customised Oracle Training Workshops and Consulting

Questions and Answers?

Presentations are available from our website:

<http://www.sagecomputing.com.au>

enquiries@sagecomputing.com.au

penny@sagecomputing.com.au