



**SAGE Computing Services**

Customised Oracle Training Workshops and Consulting



# Meet the Cost Based Optimiser in 11g

**Penny Cookson**

SAGE Computing Services

[www.sagecomputing.com.au](http://www.sagecomputing.com.au)

[penny@sagecomputing.com.au](mailto:penny@sagecomputing.com.au)



# SAGE Computing Services

Customised Oracle Training Workshops and Consulting

***Penny Cookson***

***Managing Director and Principal Consultant***

***Working with Oracle products since 1987***

***Oracle Magazine Educator of the Year 2004***



*[www.sagecomputing.com.au](http://www.sagecomputing.com.au)*

*[penny@sagecomputing.com.au](mailto:penny@sagecomputing.com.au)*



# A Characteristic Problem

I haven't changed anything

Its really slow this morning

I did the same thing yesterday and it was fine

Actually its OK now

No its not

Thank you so much you've fixed it (I haven't done anything)

# Oracle Version < 9

```
SELECT COUNT(l.quantity)
FROM bookings_skew l
WHERE resource_code = :v1;
```



'BRLG'



'PC1'



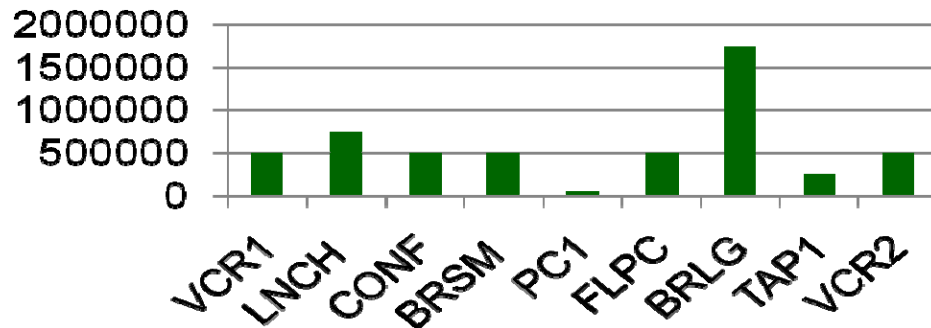
How many  
rows do I  
expect?

Shared Pool

```
SELECT COUNT(l.quantity)
FROM bookings_skew_large l
WHERE resource_code = :v1;
```

FULL SCAN

<Version 9 database  
No bind peeking



# Oracle Version < 9

```
SELECT COUNT(l.quantity)
FROM bookings_skew l
WHERE resource_code = :v1;
```



'BRLG'



'PC1'



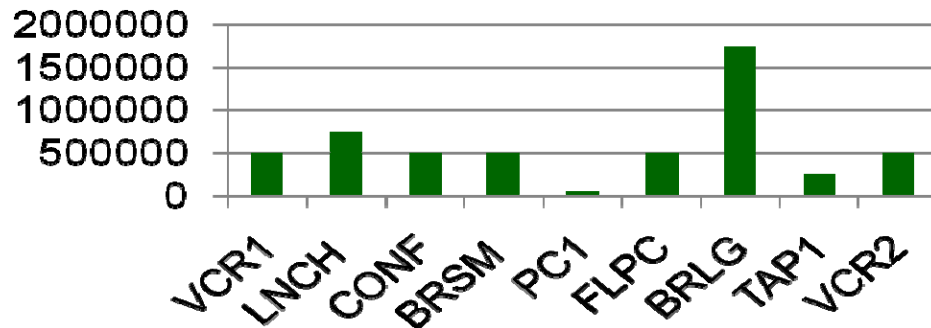
How many  
rows do I  
expect?

Shared Pool

```
SELECT COUNT(l.quantity)
FROM bookings_skew_large l
WHERE resource_code = :v1;
```

FULL SCAN

<Version 9 database  
No bind peeking



# What is Bind Peeking?

```
SELECT COUNT(l.quantity)
FROM bookings_skew l
WHERE resource_code = :v1;
```



'PC1'



'BRLG'

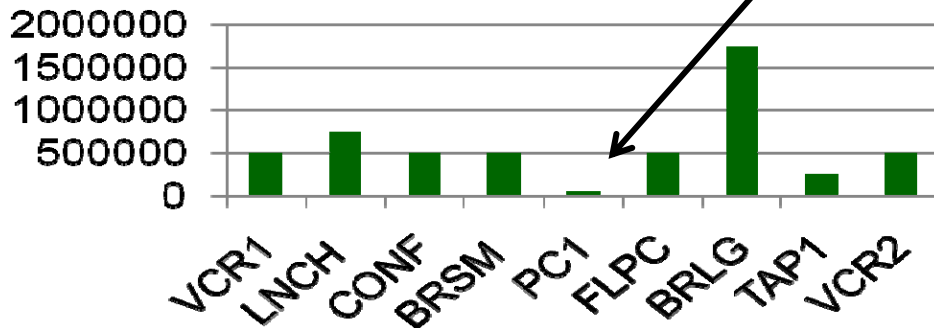


How many  
rows do I  
expect?

Shared Pool

```
SELECT COUNT(l.quantity)
FROM bookings_skew l
WHERE resource_code = :v1;
```

INDEXED  
ACCESS



>=Version 9 database  
Bind peeking



# Histogram – Minority First

\*\*\*\*\*

## SINGLE TABLE ACCESS PATH

Column (#3): RESOURCE\_CODE(VARCHAR2)

AvgLen: 5.00 NDV: 9 Nulls: 0 Density: 9.0892e-008

**Histogram:** Freq #Bkts: 9 UncompBkts: 5966 EndPtVals: 9

Table: BOOKINGS\_SKEW Alias: L

Card: **Original: 5464800 Rounded: 43968** Computed: 43967.55 Non Adjusted: 43967.55

Access Path: TableScan

Cost: 7693.35 Resp: 7693.35 Degree: 0

Cost\_io: 7426.00 Cost\_cpu: 1555877511

Resp\_io: 7426.00 Resp\_cpu: 1555877511

Access Path: index (AllEqRange)

Index: BK\_RESSKEW

resc\_io: 2399.00 resc\_cpu: 37397785

ix\_sel: 0.0080456 ix\_sel\_with\_filters: 0.0080456

Cost: 2405.43 Resp: 2405.43 Degree: 1

Best:: AccessPath: IndexRange Index: BK\_RESSKEW

Cost: 2405.43 Degree: 1 Resp: 2405.43 Card: 43967.55 Bytes: 0

\*\*\*\*\*

RESO	COUNT ( * )
----	-----
VCR1	495711
CONF	495720
LNCH	743576
BRSM	743583
PC1	47858
FLPC	495720
BRLG	1739277
TAP1	247864
VCR2	495715





# What is the CBO good at in 10g?

Data	Condition	Literal/Bind Var	Histogram	
Even Distribution	Equality	Literal	N/A	
Even Distribution	Equality	Bind	N/A	
Skewed	Equality	Literal	NO	
Skewed	Equality	Literal	YES	
Skewed	Equality	Bind	NO	
Skewed	Equality	Bind	YES	
Even Distribution	Range	Bind	N/A	

# Bind Peeking + Adaptive Cursors in 11g

Statements with bind variables (+histograms) are bind sensitive

**DEMO**

**Adaptive Cursors**

New values will use a plan for the appropriate selectivity range

# Adaptive Cursors Views

## V\$SQL

SQL_ID	CHILD_NUMBER	IS_BIND_SENSITIVE	IS_BIND_AWARE
7bmrs67hubufy	0	Y	N
7bmrs67hubufy	1	Y	Y
7bmrs67hubufy	2	Y	Y
7bmrs67hubufy	3	Y	Y

## V\$SQL\_CS\_SELECTIVITY

SQL_ID	CHILD_NUMBER	SUBSTR(PREDICATE,1,10)	RANGE_ID	LOW	HIGH
7bmrs67hubufy	1	=V1	0	0.008765	0.010712
7bmrs67hubufy	2	=V1	0	0.004382	0.010712
7bmrs67hubufy	3	=V1	0	0.004382	0.096808



# Adaptive Cursors Views

## V\$SQL\_CS\_HISTOGRAM

SQL_ID	CHILD_NUMBER	BUCKET_ID	COUNT
7bmrs67hubufy	0	1	1
7bmrs67hubufy	0	0	0
7bmrs67hubufy	0	2	1
7bmrs67hubufy	1	1	1
7bmrs67hubufy	1	0	0
7bmrs67hubufy	1	2	0
7bmrs67hubufy	2	1	0
7bmrs67hubufy	2	0	1
7bmrs67hubufy	2	2	0
7bmrs67hubufy	3	1	1
7bmrs67hubufy	3	0	11
7bmrs67hubufy	3	2	1

## V\$SQL\_CS\_STATISTICS

SQL_ID	CHILD_NUMBER	BIND_SET_HASH_VALUE	PEEKED	EXECUTIONS	ROWS_PROCESSED	BUFFER_GETS
7bmrs67hubufy	0	2982103524	Y	1	1991053	35333
7bmrs67hubufy	1	1466228028	Y	1	145519	7058
7bmrs67hubufy	2	778730927	Y	1	1	3
7bmrs67hubufy	3	2421911073	N	1	299998	4784
7bmrs67hubufy	3	3254803217	Y	1	1490902	34471



# PLSQL has soft parse avoidance

## No default adaptive cursor functionality

Implicit cursor

Explicit cursor

Native dynamic SQL

## Default adaptive cursor functionality

Ref cursor

`Session_cached_cursors = 0`



# Adaptive Cursors Functionality

## Adaptive Cursors 11.1.0.6

Bind variable with Equality and Histogram

No

# DEMO

Adap

Bin

Re

# BIND\_AWARE Hint

Does not support LIKE

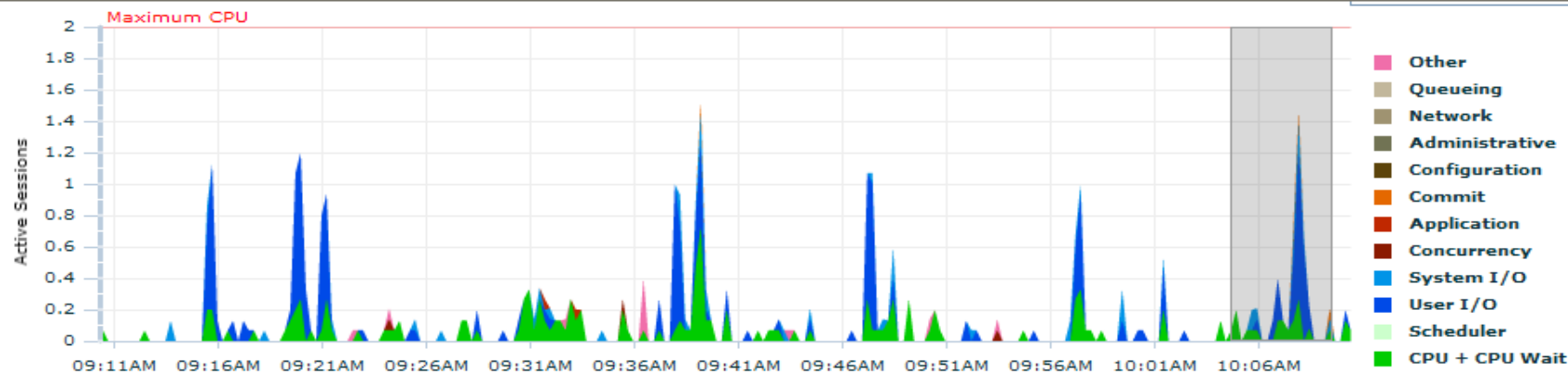
```
/*+ BIND_AWARE */
```



## So where are we now?

**Our own code – works properly first time with hint in SQL and PL/SQL**

**Packages – will still get it wrong once  
and won't use adaptive cursors for PL/SQL  
at all unless we set session cached cursors =0**



### Detail for Selected 5 Minute Interval

Start Time 7/08/2010 10:04:41 AM GMT+08:00

Run ASH Report

#### Top SQL

Actions

Select All | Select None

Select	Activity (%)	SQL ID	SQL Type
<input checked="" type="checkbox"/>	35.00	7bmrs67hubufy	SELECT
<input checked="" type="checkbox"/>	30.00	7bmrs67hubufy	SELECT
<input type="checkbox"/>	3.33	fndjrj10u6q7d	SELECT
<input type="checkbox"/>	1.67	6129566gyvx21	SELECT
<input type="checkbox"/>	1.67	cyzum8t19p208	SELECT
<input type="checkbox"/>	1.67	5955d0xup873q	SELECT
<input type="checkbox"/>	1.67	fvcwrxs43pz9	SELECT
<input type="checkbox"/>	1.67	5h0z49srcq7hr	SELECT
<input type="checkbox"/>	1.67	b6sgjk84vcnmp	SELECT

#### Top Sessions

View

Activity (%)	Session ID	User Name	Program
46.99	70	TRAIN1	sqlplus.exe
9.64	6	SYS	ORACLE.EXE (CKPT)
7.23	2	SYS	ORACLE.EXE (GEN0)
6.02	68	SYSMAN	OMS
4.82	55	SYS	ORACLE.EXE (LGWR)
3.61	21	DBSNMP	OMS
3.61	20	SYSMAN	OMS
3.61	69	SYS	OMS
2.41	27	SYS	ORACLE.EXE (J000)
2.41	58	SYSMAN	OMS

Total Sample Count: 83





```
SELECT COUNT(i.quantity)
FROM train1.bookings_large l
WHERE resource_code = :v1
```

## Details

Select the plan hash value to see the details below.

Plan Hash Value

1457437069

There are multiple plans found for this SQL statement.

[Statistics](#)

**Activity**

[Plan](#)

[Plan Control](#)

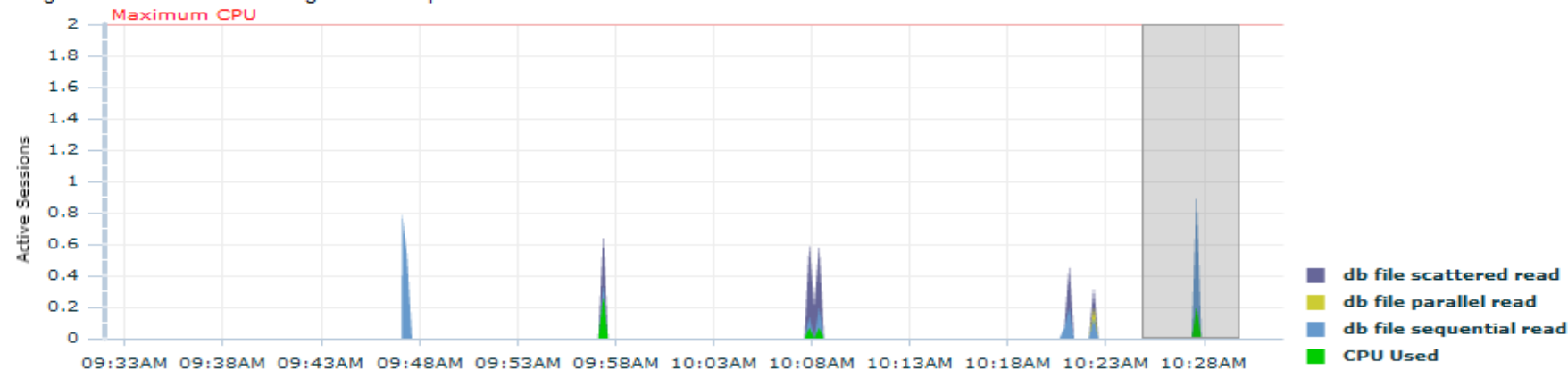
[Tuning History](#)

All  
927983165  
1457437069

[ring](#)

## Summary

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



## Detail for Selected 5 Minute Interval

Start Time 7/08/2010 10:24:52

[Run AWR SQL Report](#)

[Run ASH Report](#)

Activity (%)	SID	QC SID	User	Program	Service	Plan Hash Value
100.00	19		TRAIN1	sqlplus.exe	ora11gr2.sagecomputing.com.au	1457437069

## Details

Select the plan hash value to see the details below.

Plan Hash Value 1457437069

There are multiple plans found for this SQL statement.

[Statistics](#)[Activity](#)[Plan](#)[Plan Control](#)[Tuning History](#)[SQL Monitoring](#)

Data Source Cursor Cache

Capture Time 7/08/2010 10:14:39 (UTC+08:00)

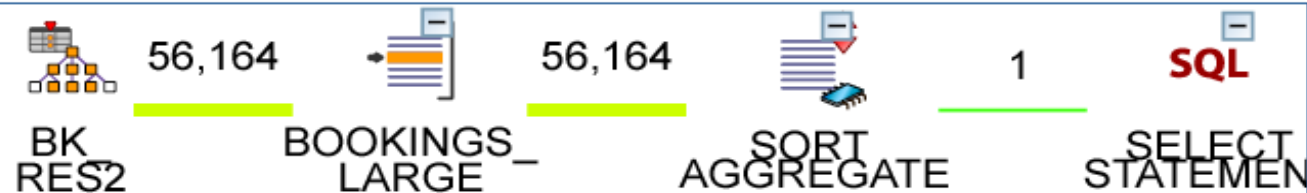
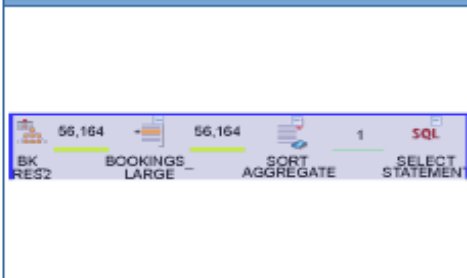
Parsing Schema TRAIN1

Optimizer Mode ALL\_ROWS

Additional Information

View ☒ Graph ☐ Table

## Overview



## Selection Details

Nothing Selected



## SQL Details: 7bmrs67hubufy

Switch to SQL ID 

Go

View Data

Real Time: Manual Refresh



Refresh

SQL Worksheet

Schedule SQL Tuning Advisor

SQL Repair Adv

Text



```
SELECT COUNT(l.quantity)
FROM train1.bookings_large l
WHERE resource_code = :v1
```

## Details

Select the plan hash value to see the details below.

Plan Hash Value



There are multiple plans found for this SQL statement.

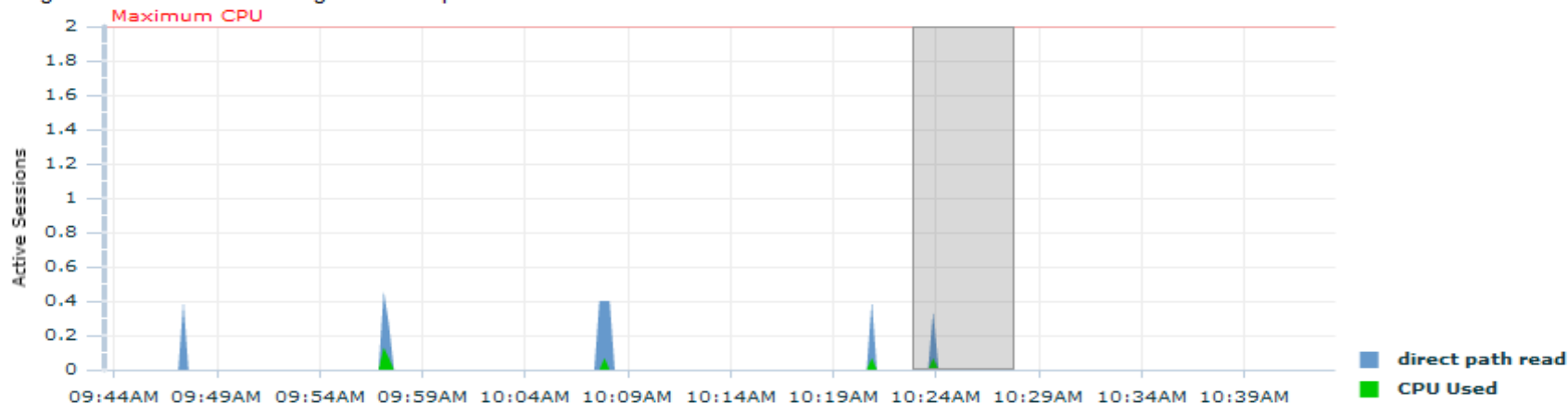
Statistics

Activity

PlanPlan ControlTuning HistorySQL Monitoring

## Summary

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





## SQL Details: 7bmrs67hubufy

Switch to SQL ID

Go

View Data

Real Time: Manual Refresh



Refresh

SQL Worksheet

Schedule SQL Tuning Advisor

SQL Repair Advisor

Text



```
SELECT COUNT(1.quantity)
FROM train1.bookings_large l
WHERE resource_code = :v1
```

## Details

Select the plan hash value to see the details below.

Plan Hash Value

927983165



There are multiple plans found for this SQL statement.

[Statistics](#)[Activity](#)[Plan](#)[Plan Control](#)[Tuning History](#)[SQL Monitoring](#)Data Source [Cursor Cache](#)

Capture Time 7/08/2010 10:33:52 (UTC+08:00)

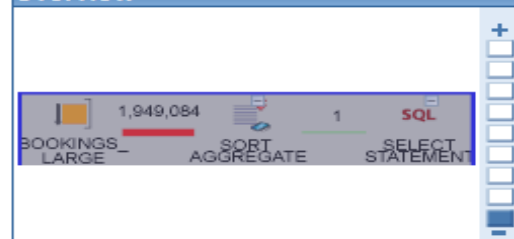
Parsing Schema TRAIN1

Optimizer Mode ALL\_ROWS

Additional Information

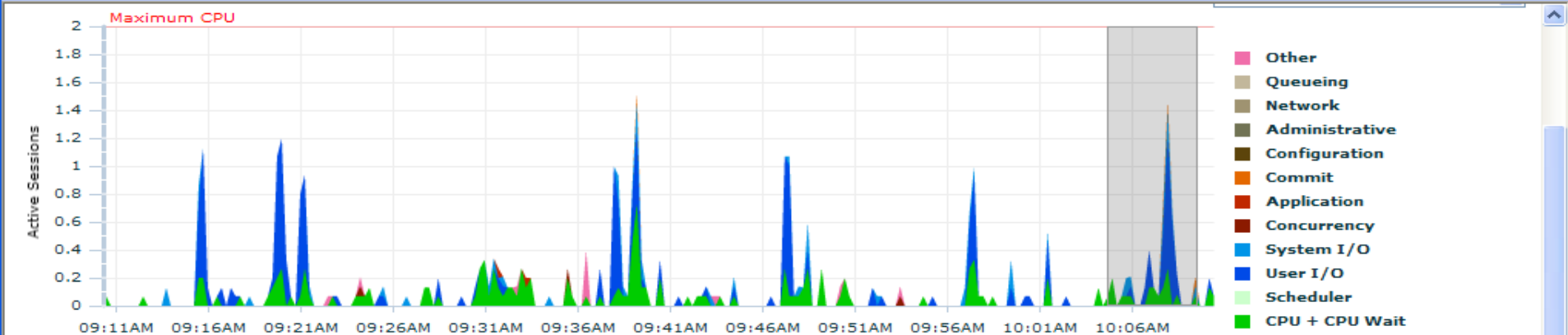
View ☒ Graph ☐ Table

## Overview



## Selection Details





### Detail for Selected 5 Minute Interval

Start Time 7/08/2010 10:04:41 AM GMT+08:00

Run ASH Report

#### Top SQL

Actions Schedule SQL Tuning Advisor Go

Select All | Select None

Select	Activity (%)	SQL ID	SQL Type
<input checked="" type="checkbox"/>	35.00	7bmrs67hubufy	SELECT
<input checked="" type="checkbox"/>	30.00	7bmrs67hubufy	SELECT
<input type="checkbox"/>	3.33	fndjrj10u6q7d	SELECT
<input type="checkbox"/>	1.67	6129566gyvx21	SELECT
<input type="checkbox"/>	1.67	cyzum8t19p208	SELECT
<input type="checkbox"/>	1.67	5955d0xup873q	SELECT
<input type="checkbox"/>	1.67	fvcwrxs43pz9	SELECT
<input type="checkbox"/>	1.67	5h0z49srcq7hr	SELECT
<input type="checkbox"/>	1.67	b6sgjk84vcnmp	SELECT

#### Top Sessions

View Top Sessions

Activity (%)	Session ID	User Name	Program
46.99	70	TRAIN1	sqlplus.exe
9.64	6	SYS	ORACLE.EXE (CKPT)
7.23	2	SYS	ORACLE.EXE (GEN0)
6.02	68	SYSMAN	OMS
4.82	55	SYS	ORACLE.EXE (LGWR)
3.61	21	DBSNMP	OMS
3.61	20	SYSMAN	OMS
3.61	69	SYS	OMS
2.41	27	SYS	ORACLE.EXE (J000)
2.41	58	SYSMAN	OMS

Total Sample Count: 83



Cancel

Submit

Specify the following parameters to schedule a job to run the SQL Tuning Advisor.

\* Name

Description

\* SQL Tuning Set



SQL Tuning Set Description **Automatically generated by Top SQL**

SQL Statements Counts 2

### ▶ SQL Statements

### Scope

Total Time Limit  
(minutes)

Scope of Analysis ☐ Limited

The analysis is done without SQL Profile recommendation and takes about 1 second per statement.

☒ Comprehensive

This analysis includes SQL Profile recommendation, but may take a long time.

Time Limit per Statement (minutes)

### Schedule

Time Zone

☒ Immediately

☐ Later

Date

(example: 7/08/2010)



Time    ☒ AM ☐ PM

Cancel

Submit

Database Instance: ora11gr2.sagecomputing.com.au > [Advisor Central](#) > [SQL Tuning Summary:SQL\\_TUNING\\_1281147098609](#) >  
[SQL Tuning Details:SQL\\_TUNING\\_1281147098609](#) >

Logged in As SYS

## Recommendations for SQL ID:7bmrs67hubufy

[Return](#)


Only one recommendation should be implemented.

### SQL Text

[SELECT COUNT\(l.quantity\) FROM train1.bookings\\_large l WHERE resource\\_code = :v1](#)

### Select Recommendation

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

Select	Type	Findings	Recommendations	Rationale	Benefit (%)	Other Statistics	New Explain Plan	Compare Explain Plans
<input checked="" type="radio"/>	Alternative Plans	Some alternative execution plans for this statement were found by searching the system's real-time and historical performance data.	The Original Plan appears to have the best performance, based on the elapsed time per execution. However, if you know that one alternative plan is better than the Original Plan, you can create a SQL plan baseline for it. This will instruct the Oracle optimizer to pick it over any other choices in the future.	Creating a plan baseline for the plan with the best elapsed time will prevent the Oracle optimizer from selecting a plan with worse performance.				

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

[Database Instance: ora11gr2.sagecomputing.com.au](#) > [Advisor Central](#) > [SQL Tuning Summary:SQL\\_TUNING\\_1281148518078](#) >[SQL Tuning Details:SQL\\_TUNING\\_1281148518078](#) >

Logged in As SYS

## Recommendations for SQL ID:7bmrs67hubufy

[Return](#)

Only one recommendation should be implemented.

### SQL Text

```
SELECT COUNT(l.quantity) FROM train1.bookings_large l WHERE resource_code = :v1
```

### Select Recommendation

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

Select	Type	Findings	Recommendations	Rationale	(%) Statistics	Plan	Compare Explain Plans
<input checked="" type="radio"/>	Alternative Plans	Some alternative execution plans for this statement were found by searching the system's real-time and historical performance data.	Consider creating a SQL plan baseline for the plan with the best average elapsed time.	Execution statistics from the cursor cache show an average elapsed time of 4.971s for the recommended plan, compared to 6.86s for the original plan. Creating a plan baseline for the plan with the best elapsed time will prevent the Oracle optimizer from selecting a plan with worse performance.			

Maybe you should buy an Exadata box

[Return](#)





# Bind Peeking + Adaptive Cursors Summary

## Your code

Use the `BIND_AWARE` hint (SQL and PL/SQL) for skewed data

Use ref cursors (if hints are not allowed)

## Packages

PL/SQL – set `session_cached_cursors = 0` (temporarily)

Minimise statement invalidations

Consider running the statement with minority and majority value on start up

?manually hack an outline/profile



# Adaptive Cursors What we Really Need

## Adaptive Cursors Persistence

Once a statement has been bind aware it knows next time its parsed

?SQL \*Profile to indicate bind aware



# Try to get rid of your hints

## `_optimizer_ignore_hints`

Create additional statistics

Set `_optimizer_ignore_hints` to TRUE

Test the app

If it runs OK remove your hints

# DEMO



## 58 New Hints

```
SELECT name, inverse, sql_feature, version  
FROM v$sql_hint  
WHERE version like '11%'  
ORDER BY version desc, name
```

[New Hints](#)



## New Hints

IGNORE\_ROW\_ON\_DUPKEY\_INDEX

This has nothing to do with optimisation I just like it

# DEMO

Document - WordPad

File Edit View Insert Format Help

SQL Monitoring Report

SQL Text

```
select /*+ MONITOR */ count(comments) from bookings
```

Global Information

```
Status          : DONE (ALL ROWS)
Instance ID     : 1
Session        : TRAIN1 (64:612)
SQL ID         : 6hxa9n8nytp4u
SQL Execution ID : 16777217
Execution Started : 08/09/2010 15:52:43
First Refresh Time : 08/09/2010 15:52:43
Last Refresh Time : 08/09/2010 15:52:43
Duration       : .000169s
Module/Action  : SQL Developer/-
Service        : SYS$USERS
Program       : SQL Developer
Fetch Calls    : 1
```

Global Stats

Elapsed Time(s)	Other Waits(s)	Fetch Calls	Buffer Gets
0.00	0.00	1	7

SQL Plan Monitoring Details (Plan Hash Value=21997667)

Id	Operation	Name	Rows (Estim)	Cost	Time Active(s)	Start Active	Execs	Rows (Actual)	Activity (\$)	Activity Detail (# samples)
0	SELECT STATEMENT				1	+0	1	1		
1	SORT AGGREGATE		1		1	+0	1	1		
2	TABLE ACCESS FULL	BOOKINGS	22	3	1	+0	1	22		

For Help, press F1

from dual;



# Much More Query Transformation

```
SELECT e.event_no, e.start_date, sum(cost) totcost  
FROM    events_large e, bookings_large b  
WHERE   e.event_no = b.event_no  
GROUP BY e.event_no, e.start_date
```

```
alter session set tracefile_identifier = Penny  
  
alter session set events '10053 trace name context forever'  
  
explain plan for  
SELECT e.event_no, e.start_date, sum(cost) totcost  
FROM    events_large e, bookings_large b  
WHERE   e.event_no = b.event_no  
GROUP BY e.event_no, e.start_date;  
  
alter session set events '10053 trace name context off'
```



# Much More Query Transformation

```
SELECT e.event_no, e.start_date, sum(cost) totcost
FROM   events_large e, bookings_large b
WHERE  e.event_no = b.event_no
GROUP BY e.event_no, e.start_date
```

A screenshot of a Windows WordPad window titled "ora11gr2\_trace10053.trc - WordPad". The window displays the "Query Block Registry" section of an Oracle trace file. The text shows various query blocks with their origins and list numbers, including blocks for parsing, table changes, and predicate pushing. At the bottom, there is a call to the "compile" function with memory allocation details.

```
Query Block Registry:
SEL$1 0x0 (PARSER)
  SEL$54B5B641 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST LIST 3)
  SEL$54B5B641 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST LIST 3)
  SEL$706665FA 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST 2)
  SEL$706665FA 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST 2)
  SEL$38F5D95B 0x0 (QUERY BLOCK TABLES CHANGED SEL$1)
    SEL$137A03FC 0x0 (SPLIT/MERGE QUERY BLOCKS SEL$38F5D95B)
  SEL$7C398D44 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST 5) [FINAL]
    SEL$7A931295 0x0 (PUSHED PREDICATE SEL$7D2C682D; SEL$7C398D44; "VW_GBC_5"@SEL$F486F43F" 1)
  SEL$7C398D44 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST 5) [FINAL]
  ...
  SEL$F486F43F 0x0 (QUERY BLOCK TABLES CHANGED SEL$1)
    SEL$1DBAD500 0x0 (QUERY BLOCK TABLES CHANGED SEL$F486F43F)|
    SEL$6D7A4A3D 0x0 (SPLIT/MERGE QUERY BLOCKS SEL$1DBAD500)
    SEL$7D2C682D 0x0 (SPLIT/MERGE QUERY BLOCKS SEL$F486F43F) [FINAL]
    SEL$7A931295 0x0 (PUSHED PREDICATE SEL$7D2C682D; SEL$7C398D44; "VW_GBC_5"@SEL$F486F43F" 1)

:
  call(in-use=85904, alloc=114672), compile(in-use=364968, alloc=472468), execution(in-use=498372, alloc=499124)
```

For Help, press F1



Oracle SQL Developer : train1\_ora10

File Edit View Navigate Run Versioning Tools Help

11gr2\_train1 x

train1\_ora10 x

EVENTS\_LARGE x

0 seconds

train1\_ora10

1 SELECT e.event\_no, e.start\_date, sum(cost) totcost

2 FROM events\_large e, bookings\_large b

3 where e.event\_no = b.event\_no

4 group by e.event\_no, e.start\_date

5

10G – JOIN before the GROUP BY

Explain Plan x

0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALITY
SELECT STATEMENT			31665	843726
HASH		GROUP BY	31665	843726
HASH JOIN			14060	5777591
Access Predicates				
E.EVENT_NO=B.EVENT_NO				
TABLE ACCESS	EVENTS_LARGE	FULL	225	99827
TABLE ACCESS	BOOKINGS_LARGE	FULL	8044	5777591

Data Editor - Log x

Dbms Output x

SQL History x

Line 5 Column 1 | Insert | Modified | Windows: CR/LF Editing

```

----- waited -----
SQL*Net message to client          1          0.00          0.00
SQL*Net message from client        1          0.00          0.00
*****

```

```

SELECT e.event_no, e.start_date, sum(cost) totcost
FROM   events_large e, bookings_large b
where  e.event_no = b.event_no
group by e.event_no, e.start_date

```

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	6668	18.43	22.06	28499	36330	0	100001
total	6670	18.43	22.06	28499	36330	0	100001

```

Misses in library cache during parse: 1
Optimizer mode: ALL_ROWS
Parsing user id: 87

```

Rows	Row Source Operation
100001	HASH GROUP BY (cr=36330 pr=28499 pw=0 time=21681647 us)
5767168	HASH JOIN (cr=36330 pr=28499 pw=0 time=11813763 us)
100322	TABLE ACCESS FULL EVENTS_LARGE (cr=976 pr=0 pw=0 time=119 us)
5767168	TABLE ACCESS FULL BOOKINGS_LARGE (cr=35354 pr=28499 pw=0 time=5812854 us)

Elapsed times include waiting on following events:

Event waited on	Times waited	Max. Wait	Total Waited
SQL*Net message to client	6668	0.00	0.00
db file scattered read	2416	0.04	3.69
db file sequential read	113	0.02	0.12

Oracle SQL Developer : 11gr2\_train1

File Edit View Navigate Run Versioning Tools Help

11gr2\_train1 x train1\_oracle10 x EVENTS\_LARGE x

0.016 seconds

ask Tom

```
1 SELECT e.event_no, e.start_date, sum(cost) totcost
2 FROM   events_large e, bookings_large b
3 where  e.event_no = b.event_no
4 group by e.event_no, e.start_date
5
6
```

## 11G – GROUP BY before the JOIN

Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Explain Plan x

0.016 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALITY
SELECT STATEMENT			19036	100322
HASH		GROUP BY	19036	100322
HASH JOIN			18176	100322
Access Predicates E.EVENT_NO=ITEM_1				
TABLE ACCESS	EVENTS_LARGE	FULL	273	100322
VIEW	VW_GBC_5		17637	100792
HASH		GROUP BY	17637	100792
TABLE ACCESS	BOOKINGS_LARGE	FULL	9836	5767168

Data Editor - Log x Dbms Output x SQL History x

train1\_oracle10

Line 4 Column 34 | Insert | Modified | Windows: CR/LF Editing

\*\*\*\*\*

```
SELECT e.event_no, e.start_date, sum(cost) totcost
FROM   events_large e, bookings_large b
where  e.event_no = b.event_no
group by e.event_no, e.start_date
```

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	6668	4.03	6.26	35213	36203	0	100000
total	6670	4.04	6.27	35213	36203	0	100000

Misses in library cache during parse: 1

Optimizer mode: ALL\_ROWS

Parsing user id: 92

Rows Row Source Operation

```
100000 HASH GROUP BY (cr=36203 pr=35213 pw=0 time=66708 us cost=19036
size=3109982 card=100322)
100000 HASH JOIN (cr=36203 pr=35213 pw=0 time=301403 us cost=18176
size=3109982 card=100322)
100322 TABLE ACCESS FULL EVENTS_LARGE (cr=979 pr=0 pw=0 time=108254 us
cost=273 size=1304186 card=100322)
100000 VIEW VW_GBC_5 (cr=35224 pr=35213 pw=0 time=128003 us cost=17637
size=1814256 card=100792)
100000 HASH GROUP BY (cr=35224 pr=35213 pw=0 time=57288 us cost=17637
size=806336 card=100792)
5767168 TABLE ACCESS FULL BOOKINGS_LARGE (cr=35224 pr=35213 pw=0
time=7637118 us cost=9836 size=46137344 card=5767168)
```

Elapsed times include waiting on following events:

Event waited on	Times waited	Max. wait	Total waited
SQL*Net message to client	6668	0.00	0.01
direct path read	206	0.06	2.05
asynch descriptor resize	4	0.00	0.00
SQL*Net message from client	6668	15.08	45.14



Registered qb: SEL\$1 0x2a656ac0 (COPY SEL\$1)

# QUERY BLOCK SIGNATURE

signature(): NULL

\*\*\*\*\*

Cost-Based Group-By/Distinct Placement

\*\*\*\*\*

GBP/DP: Checking validity of GBP/DP for query block SEL\$1 (#1)

GBP: Checking validity of group-by placement for query block SEL\$1 (#1)

GBP: Using search type: exhaustive

GBP: Considering group-by placement on query block SEL\$1 (#1)

GBP: Starting iteration 1, state space = (1,2) : (0,0)

GBP: Original query

\*\*\*\*\* UNPARSED QUERY IS \*\*\*\*\*

SELECT "E"."EVENT\_NO" "EVENT\_NO", "E"."START\_DATE" "START\_DATE", SUM("B"."COST") "TOTCOST" FROM "TRAIN1"."EVENTS\_LARGE"

"E"."TRAIN1"."BOOKINGS\_LARGE" "B" WHERE "E"."EVENT\_NO"="B"."EVENT\_NO" GROUP BY "E"."EVENT\_NO", "E"."START\_DATE"

FPD: Considering simple filter push in query block SEL\$1 (#1)

"E"."EVENT\_NO"="B"."EVENT\_NO"

try to generate transitive predicate from check constraints for query block SEL\$1 (#1)

finally: "E"."EVENT\_NO"="B"."EVENT\_NO"

GBP: Costing transformed query.

CBQT: Looking for cost annotations for query block SEL\$1, key = SEL\$1\_00000000\_0

CBQT: Could not find stored cost annotations.

kkqbbc: optimizing query block SEL\$1 (#1)

call(in-use=4440, alloc=16360), compile(in-use=141756, alloc=158708), execution(in-use=176260, alloc=179748)

kkqbbc-subheap (create addr=0x0E42EF5C)

\*\*\*\*\*

QUERY BLOCK TEXT

\*\*\*\*\*

Not available.

# QUERY BLOCK SIGNATURE

signature (optimizer): qb\_name=SEL\$1 nbfros=2 flg=0

fro(0): flg=0 objn=74856 hint\_alias="B"@SEL\$1"

fro(1): flg=0 objn=74848 hint\_alias="E"@SEL\$1"

# SYSTEM STATISTICS INFORMATION

Using NOWORKLOAD Stats

CPUSPEEDNW: 1684 millions instructions/sec (default is 100)

IOTFRSPEED: 4096 bytes per millisecond (default is 4096)

IOSEEKTIM: 10 milliseconds (default is 10)

MBRC: -1 blocks (default is 8)

\*\*\*\*\*

```
ora11gr2_ora_3740_PENNY4.trc - Notepad
File Edit Format View Help
*****
Not available.
-----
QUERY BLOCK SIGNATURE
-----
signature (optimizer): qb_name=SEL$7C398D44 nbfros=2 flg=0
fro(0): flg=0 objn=74848 hint_alias="E"@SEL$1"
fro(1): flg=1 objn=0 hint_alias="VW_GBC_5"@SEL$F486F43F"
-----
SYSTEM STATISTICS INFORMATION
-----
Using NOWORKLOAD Stats
CPUSPEEDNW: 1684 millions instructions/sec (default is 100)
IOTFRSPEED: 4096 bytes per millisecond (default is 4096)
IOSEEKTIM: 10 milliseconds (default is 10)
MBRC: -1 blocks (default is 8)
*****
BASE STATISTICAL INFORMATION
*****
Table Stats::
Table: EVENTS_LARGE Alias: E
#Rows: 100322 #Blks: 1000 AvgRowLen: 64.00
Index Stats::
Index: EVTLG_PK Col#: 1
LVLS: 1 #LB: 188 #DK: 100322 LB/K: 1.00 DB/K: 1.00 CLUF: 956.00
*****
Table Stats::
Table: VW_GBC_5 Alias: VW_GBC_5 NO STATISTICS
Column (#1): ITEM_1C
AvgLen: 5 NDV: 100792 Nulls: 0 Density: 0.000010 Min: 201 Max: 100200
Access path analysis for VW_GBC_5
Access path analysis for EVENTS_LARGE
*****
SINGLE TABLE ACCESS PATH
Single Table Cardinality Estimation for EVENTS_LARGE[E]
Table: EVENTS_LARGE Alias: E
Card: Original: 100322.000000 Rounded: 100322 Computed: 100322.00 Non Adjusted: 100322.00
Access Path: TableScan
Cost: 273.49 Resp: 273.49 Degree: 0
Cost_io: 272.00 Cost_cpu: 30195500
Resp_io: 272.00 Resp_cpu: 30195500
Best:: AccessPath: TableScan
Cost: 273.49 Degree: 1 Resp: 273.49 Card: 100322.00 Bytes: 0
Grouping column cardinality [ EVENT_NO] 100322
Grouping column cardinality [START_DATE] 12
*****
OPTIMIZER STATISTICS AND COMPUTATIONS
```

```

fix 8355120 = enabled
fix 7176746 = enabled
fix 8442891 = enabled
fix 8373261 = enabled
fix 7679164 = enabled
fix 7670533 = enabled
fix 8408665 = enabled
fix 8491399 = enabled
fix 8348392 = enabled
fix 8348585 = enabled
fix 8508056 = enabled
fix 8335178 = enabled
fix 8515269 = enabled
fix 8247017 = enabled
fix 7325597 = enabled
fix 8531490 = enabled
fix 6163600 = enabled
fix 8589278 = disabled
fix 8557992 = enabled
fix 7556098 = enabled
fix 8580883 = enabled
fix 5892599 = disabled
fix 8609714 = enabled
fix 8514561 = enabled
fix 8619631 = disabled

```

## Query Block Registry:

```

SEL$1 0x0 (PARSER)
  SEL$54B5B641 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST LIST 3)
  SEL$54B5B641 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST LIST 3)
  SEL$706665FA 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST 2)
  SEL$706665FA 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST 2)
  SEL$38F5D95B 0x0 (QUERY BLOCK TABLES CHANGED SEL$1)
    SEL$137A03FC 0x0 (SPLIT/MERGE QUERY BLOCKS SEL$38F5D95B)
  SEL$7C398D44 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST 5) [FINAL]
    SEL$7A931295 0x0 (PUSHED PREDICATE SEL$7D2C682D; SEL$7C398D44; "vw_GBC_5"@SEL$F486F43F" 1)
  SEL$7C398D44 0x0 (UNKNOWN QUERY BLOCK ORIGIN SEL$1; SEL$1; LIST 5) [FINAL]

SEL$F486F43F 0x0 (QUERY BLOCK TABLES CHANGED SEL$1)
  SEL$1DBAD500 0x0 (QUERY BLOCK TABLES CHANGED SEL$F486F43F)
    SEL$6D7A4A3D 0x0 (SPLIT/MERGE QUERY BLOCKS SEL$1DBAD500)
  SEL$7D2C682D 0x0 (SPLIT/MERGE QUERY BLOCKS SEL$F486F43F) [FINAL]
    SEL$7A931295 0x0 (PUSHED PREDICATE SEL$7D2C682D; SEL$7C398D44; "vw_GBC_5"@SEL$F486F43F" 1)

:
  call(in-use=85904, alloc=114672), compile(in-use=364968, alloc=472468), execution(in-use=498372, alloc=499124)

```

End of Optimizer State Dump

Dumping Hints

```

=====
===== END SQL Statement Dump =====

```

```

SELECT e.event_no, e.start_date, sum(cost) totcost
FROM   events_large e, bookings_large b
where  e.event_no = b.event_no
group by e.event_no, e.start_date
----- Explain Plan Dump -----
----- Plan Table -----

```

```

=====
Plan Table
=====

```

Id	Operation	Name	Rows	Bytes	Cost	Time
0	SELECT STATEMENT				19K	
1	HASH GROUP BY		98K	3037K	19K	00:04:49
2	HASH JOIN		98K	3037K	18K	00:04:39
3	TABLE ACCESS FULL	EVENTS_LARGE	98K	1274K	273	00:00:04
4	VIEW	VW_GBC_5	98K	1772K	17K	00:04:32
5	HASH GROUP BY		98K	787K	17K	00:04:32
6	TABLE ACCESS FULL	BOOKINGS_LARGE	5632K	44M	9836	00:02:59

```

-----
Predicate Information:
-----

```

```

2 - access("E"."EVENT_NO"="ITEM_1")

```

```

Content of other_xml column

```

```

=====
db_version      : 11.2.0.1
parse_schema    : TRAIN1
plan_hash       : 1200697483
plan_hash_2     : 3556861701
Outline Data:
/*+
  BEGIN_OUTLINE_DATA
  IGNORE_OPTIM_EMBEDDED_HINTS
  OPTIMIZER_FEATURES_ENABLE('11.2.0.1')
  DB_VERSION('11.2.0.1')
  OPT_PARAM('optimizer_index_cost_adj' 20)
  ALL_ROWS
  OUTLINE_LEAF(@"SEL$7D2C682D")
  OUTLINE_LEAF(@"SEL$7C398D44")
  PLACE_GROUP_BY(@"SEL$1" ( "B"@"SEL$1" ) 5)
  OUTLINE(@"SEL$F486F43F")
  OUTLINE(@"SEL$1")
  FULL(@"SEL$7C398D44" "E"@"SEL$1")
  NO_ACCESS(@"SEL$7C398D44" "VW_GBC_5"@"SEL$F486F43F")
  LEADING(@"SEL$7C398D44" "E"@"SEL$1" "VW_GBC_5"@"SEL$F486F43F")
  USE_HASH(@"SEL$7C398D44" "VW_GBC_5"@"SEL$F486F43F")
  USE_HASH_AGGREGATION(@"SEL$7C398D44")
  FULL(@"SEL$7D2C682D" "B"@"SEL$1")
  USE_HASH_AGGREGATION(@"SEL$7D2C682D")
END_OUTLINE_DATA
*/

```



# 10G – NOT IN (PROMISE\_NO NULLABLE)

```
20
21 SELECT count(*)
22 FROM election_promises p
23 WHERE p.promise_no NOT IN
24 (SELECT promise_no
25 FROM fulfilled_promises
26 WHERE status = 'Y'
27 AND cost <1000000)
28
```

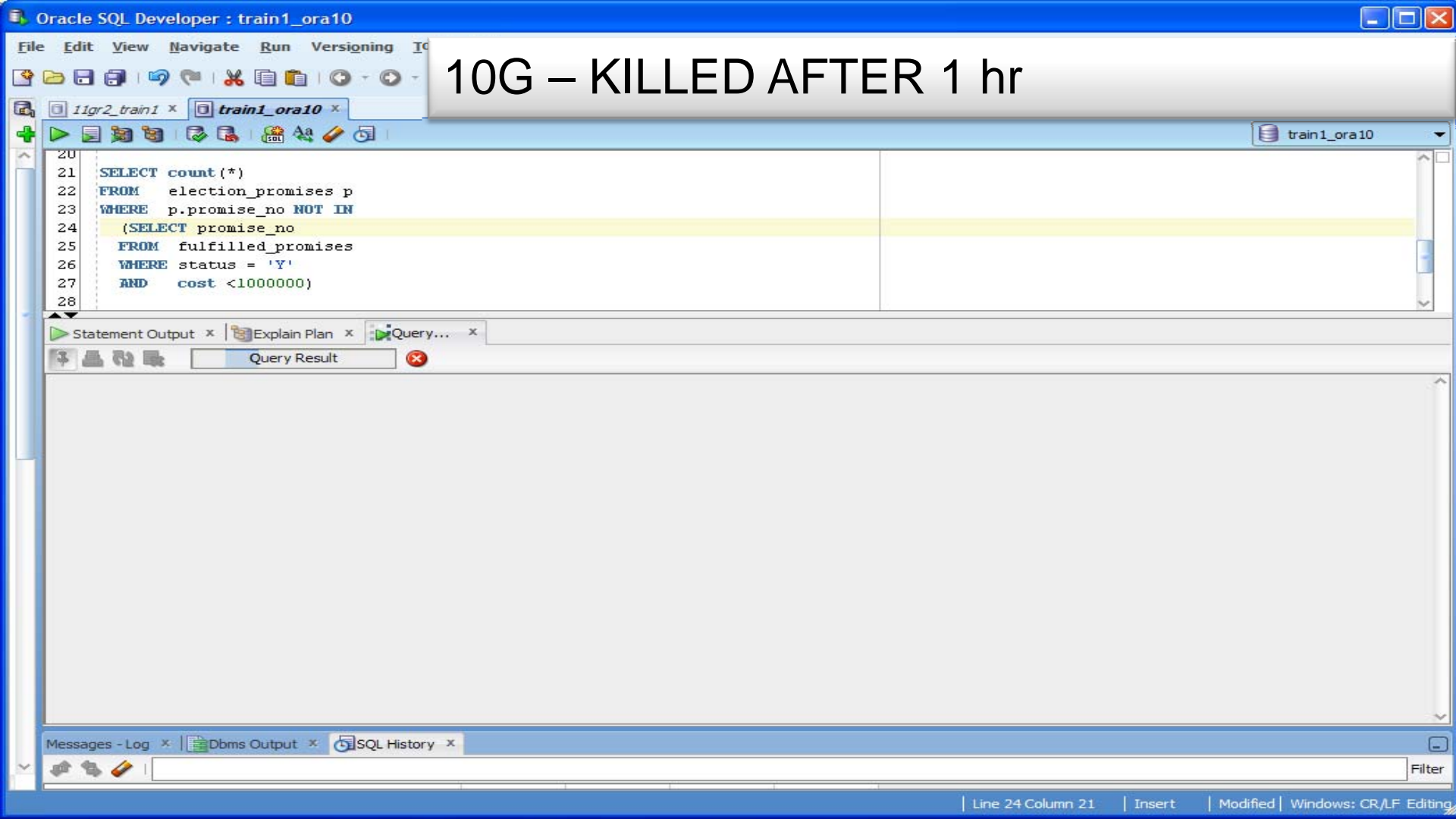
Statement Output x Explain Plan x

0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALITY
SELECT STATEMENT			29549688	1
SORT		AGGREGATE		1
FILTER				
Filter Predicates				
NOT EXISTS (SELECT /*+ */ 0 FROM FULFILLED_PROMISES FULFILL				
INDEX	ELPR_FK	FAST FULL SCAN	53	116035
TABLE ACCESS	FULFILLED_PROMISES	FULL	8149	1188703
Filter Predicates				
AND				
STATUS='Y'				
COST<1000000				
LNNVL(PROMISE_NO<>:B1)				

Messages - Log x Dbms Output x SQL History x

Filter



10G – KILLED AFTER 1 hr

```
20  
21 SELECT count(*)  
22 FROM election_promises p  
23 WHERE p.promise_no NOT IN  
24       (SELECT promise_no  
25        FROM fulfilled_promises  
26         WHERE status = 'Y'  
27         AND cost <1000000)  
28
```

Statement Output x Explain Plan x Query... x

Query Result

Messages - Log x Dbms Output x SQL History x

# 11G – Rewrite as Null Aware Anti join

```
20
21 SELECT count(*)
22 FROM   election_promises p
23 WHERE  p.promise_no NOT IN
24        (SELECT promise_no
25         FROM   fulfilled_promises
26         WHERE  status = 'Y'
27         AND    cost < 1000000)
28
```

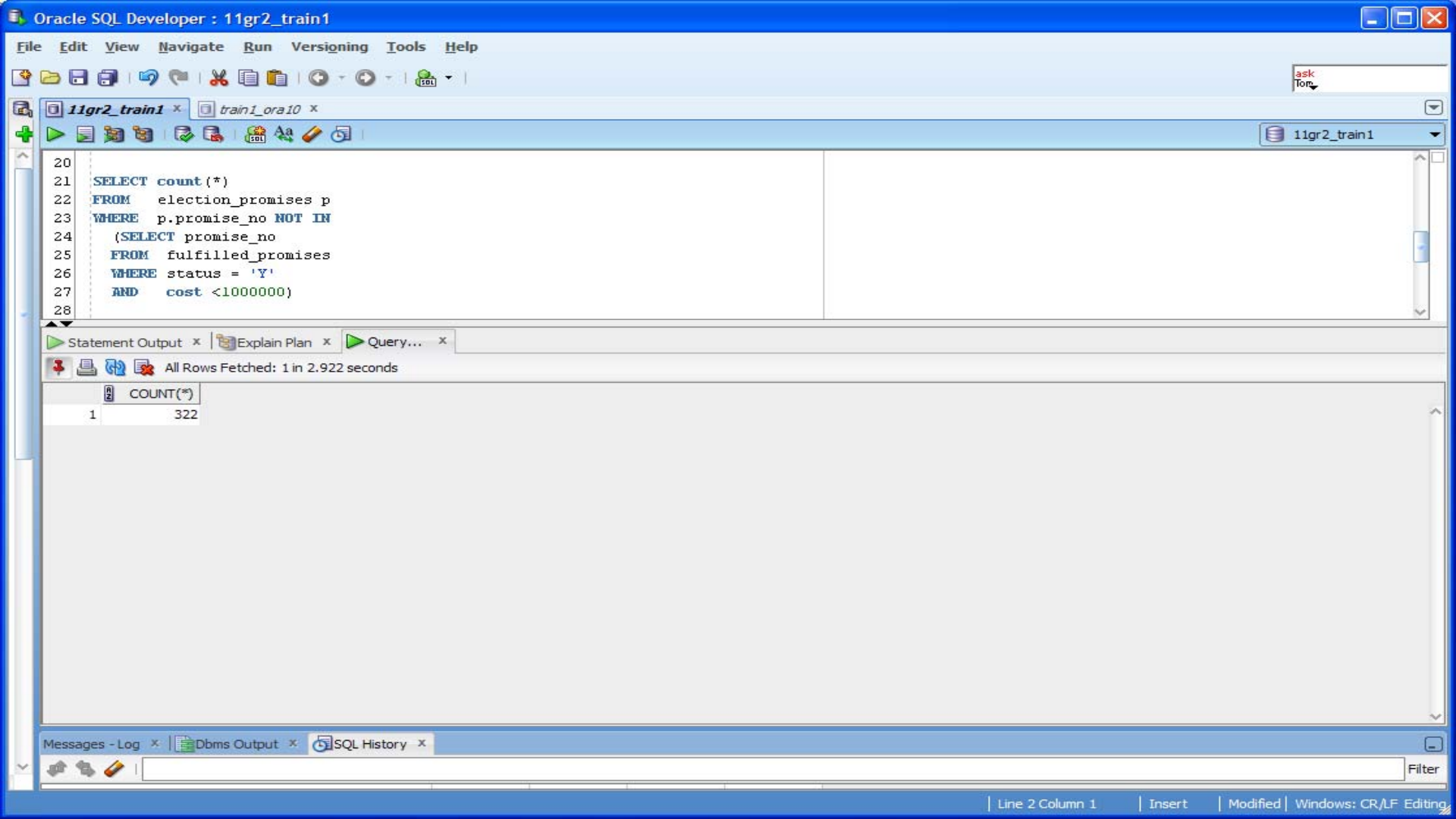
Statement Output x Explain Plan x

0.14 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALITY
SELECT STATEMENT			11966	1
SORT		AGGREGATE		1
HASH JOIN		ANTI NA	11966	102893
Access Predicates				
P.PROMISE_NO=PROMISE_NO				
INDEX	ELPR_PK	FAST FULL SCAN	63	102893
TABLE ACCESS	FULFILLED_PROMISES	FULL	9747	1069449
Filter Predicates				
AND				
STATUS='Y'				
COST < 1000000				

Messages - Log x Dbms Output x SQL History x

Filter



# 10G – Full Outer Join – default behaviour

File Edit View Navigate Run Versioning Tools

11gr2\_train1 x train1\_ora10 x

0.016 seconds

train1\_ora10

```
1 SELECT count(s.seat_no), count(s.description), count(g.preferences)
2 FROM seat s FULL OUTER JOIN green_votes g ON {s.seat_no = g.seat_no}
```

Query Result x Statement Output x Explain Plan x

0.016 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALITY
SELECT STATEMENT			29026	1
SORT		AGGREGATE		1
VIEW			29026	5765805
UNION-ALL				
HASH JOIN		OUTER	19010	5676089
Access Predicates				
S.SEAT_NO=G.SEAT_NO(+)				
TABLE ACCESS	SEAT	FULL	222	98128
TABLE ACCESS	GREEN_VOTES	FULL	8134	5765805
NESTED LOOPS		ANTI	10016	89716
TABLE ACCESS	GREEN_VOTES	FULL	8134	5765805
INDEX	ST_FK	UNIQUE SCAN	0	96601
Access Predicates				
S.SEAT_NO=G.SEAT_NO				

Messages - Log x Dbms Output x SQL History x

Filter

```
1 SELECT count(s.seat_no), count(s.description), count(g.preferences)
2 FROM seat s FULL OUTER JOIN green_votes g ON {s.seat_no = g.seat_no}
```

    All Rows Fetched: 1 in 52.734 seconds

	A2	A2	A2
	COUNT(S.SEAT_NO)	COUNT(S.DESCRPTION)	COUNT(G.PREFERENCES)
1	5767489	5767489	786432



## From version 10.2.0.3

10G – Full Outer Join – hint

```
SELECT /*+ NATIVE_FULL_OUTER_JOIN */  
      count(e.comments) nume,  
      count (b.comments) numb  
FROM   events_large e  
       FULL OUTER JOIN bookings_large b  
       ON (e.event_no = b.event_no)
```



Oracle SQL Developer : 11gr2\_train1

File Edit View Navigate Run Versioning Tools

11gr2\_train1 x train1\_ora10 x

0.015 seconds

11gr2\_train1

1  
2

SELECT count(s.seat\_no), count(s.description), count(g.preferences)  
FROM seat s FULL OUTER JOIN green\_votes g ON (s.seat\_no = g.seat\_no)

Query Result x Statement Output x Query Result 1 x Query Result 2 x Query Result 3 x Explain Plan x

0.015 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST	CARDINALITY
SELECT STATEMENT			17359	1
SORT		AGGREGATE		1
VIEW	VW_FOJ_0		17359	5767168
HASH JOIN		FULL OUTER	17359	5767168
Access Predicates S.SEAT_NO=G.SEAT_NO				
TABLE ACCESS	SEAT	FULL	270	100322
TABLE ACCESS	GREEN_VOTES	FULL	9737	5767168

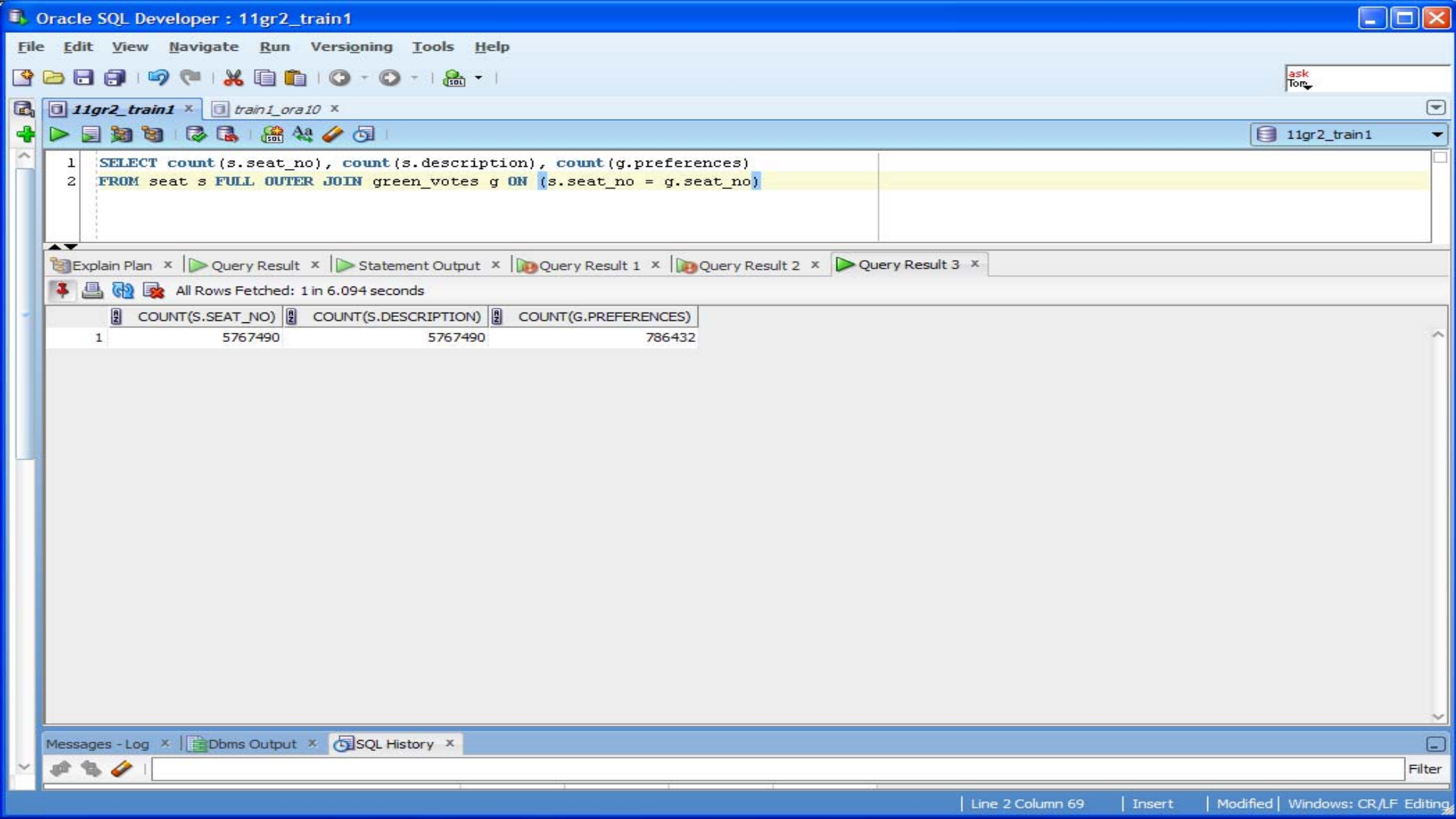
Messages - Log x Dbms Output x SQL History x

Filter

Line 2 Column 69 | Insert | Modified | Windows: CR/LF Editing

11G Native Full Outer Join





```
1 SELECT count(s.seat_no), count(s.description), count(g.preferences)
2 FROM seat s FULL OUTER JOIN green_votes g ON (s.seat_no = g.seat_no)
```

Explain Plan x Query Result x Statement Output x Query Result 1 x Query Result 2 x Query Result 3 x

All Rows Fetched: 1 in 6.094 seconds

	COUNT(S.SEAT_NO)	COUNT(S.DESCRPTION)	COUNT(G.PREFERENCES)
1	5767490	5767490	786432

Messages - Log x Dbms Output x SQL History x

Filter



# Upgrading

Check the differences in plans

```
rundiff.txt - Notepad
File Edit Format View Help

set serveroutput on
set define off

declare
    result varchar2(1000);
BEGIN
    result := dbms_xplan.diff_plan_outline(
        sql_text      => 'SELECT e.event_no, e.start_date FROM events_large e WHERE event_no
NOT IN
(SELECT event_no FROM bookings_large WHERE status = ''P'' AND cost > 100)',
        outline1       => 'OPTIMIZER_FEATURES_ENABLE(''11.1.0.7'')',
        outline2       => 'OPTIMIZER_FEATURES_ENABLE(''10.1.0.2'')';
    dbms_output.put_line('diff is '||result);
end;

URL:|
http://host.my.com:portnumber/orarep/plandiff/all?task_id=238&format=html&method=qbreg

select dbms_report.get_report(
        '/orarep/plandiff/all?task_id=238&format=text&method=qbreg')
from dual;
```



```
DBMS_REPORT.GET_REPORT('/ORAREP/PLANDIFF/ALL?TASK_ID=238&FORMAT=TEXT&METHOD=QBRE
```

```
-----
General Information
-----
```

```
Task Information:
```

```
Workload Information:
```

```
-----
Task Name       : TASK_238
Task Owner      : TRAIN1
Description     : diff_plan_outline
-----
```

```
Report Details: SQL Plan Comparison Query Block Diff
-----
```

Query Block	SQL Plan 1	SQL Plan 2	Diff
.. SEL\$5DA710D3	Yes	NA	.. SUBQUERY UNNEST

```
Plan Id       : 994
Plan Hash Value : 3319067774
-----
```

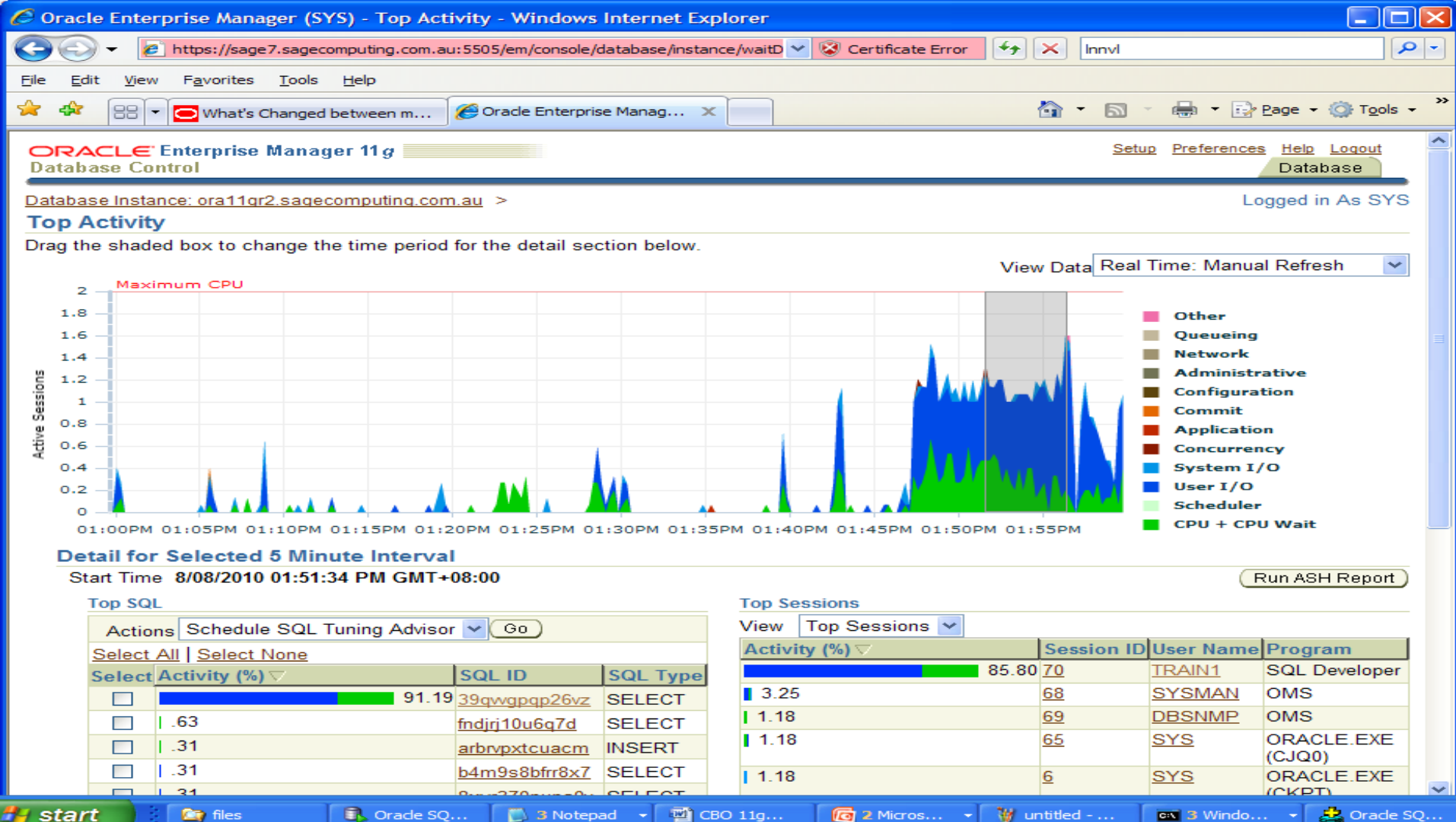
Id	Operation	Name
----	-----------	------

```
Plan Id       : 995
Plan Hash Value : 1140461147
-----
```

Id	Operation	Name
----	-----------	------

Diff	Transformation	SQL Plan 1	SQL Plan 2	Query block
*	. PARSER	-	Yes	. SEL\$2
*	.. SUBQUERY UNNEST	Yes	NA	.. SEL\$5DA710D3

Diff	Transformation	SQL Plan 1	SQL Plan 2	Query block
*	. PARSER	-	Yes	. SEL\$1
v	.. SUBQUERY UNNEST	Yes	NA	.. SEL\$5DA710D3



```

/* SQL Analyze(70,0) */
SELECT e.event_no, e.start_date
FROM events_large e
WHERE event_no NOT IN
(SELECT event_no
FROM bookings_large
WHERE status = 'P' AND cost > 100)

```

## Details

Select the plan hash value to see the details below.

Plan Hash Value

There are multiple plans found for this SQL statement.

[Statistics](#)

[Activity](#)

[Plan](#)

[Plan Control](#)

[Tuning History](#)

[SQL Monitoring](#)

## Summary

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



## Detail for Selected 5 Minute Interval

Start Time 8/08/2010 13:51:34

[Run AWR SQL Report](#)

[Run ASH Report](#)

Activity (%)	SID	QC SID	User	Program	Service	Plan Hash Value
100.00	70		TRAIN1	SQL Developer	SYSSUSERS	1140461147

# Oracle 10g – Run a whole load of random statements – all of which require parsing

## OVERALL TOTALS FOR ALL NON-RECURSIVE STATEMENTS

call	count	cpu	elapsed	disk	query	current	rows
Parse	1251	1.87	1.90	0	0	0	0
Execute	1252	0.04	0.05	0	57	0	2
Fetch	2700	25.32	25.38	0	429131	0	12400
total	5203	27.25	27.34	0	429188	0	12402

Misses in library cache during parse: 1251

Misses in library cache during execute: 1

Elapsed times include waiting on following events:

Event waited on	Times Waited	Max. Wait	Total Waited
SQL*Net message to client	4251	0.00	0.00
SQL*Net message from client	4251	0.04	1.48
SQL*Net break/reset to client	198	0.00	0.01

## OVERALL TOTALS FOR ALL RECURSIVE STATEMENTS

call	count	cpu	elapsed	disk	query	current	rows
Parse	63	0.00	0.00	0	0	0	0
Execute	165	0.01	0.01	0	0	0	0
Fetch	298	0.04	0.02	0	4363	0	220
total	526	0.06	0.04	0	4363	0	220

Misses in library cache during parse: 15

Misses in library cache during execute: 15

# Oracle 11g – Run a whole load of random statements – all of which require parsing

**So Tuning the Shared Pool becomes more important, but overall the 11g CBO is pretty smart**

OVERALL TOTALS FOR ALL NON-RECURSIVE STATEMENTS

call	count
Parse	1251
Execute	1252
Fetch	2700
total	5203

Misses in library cache during parse: 115

Elapsed times in seconds

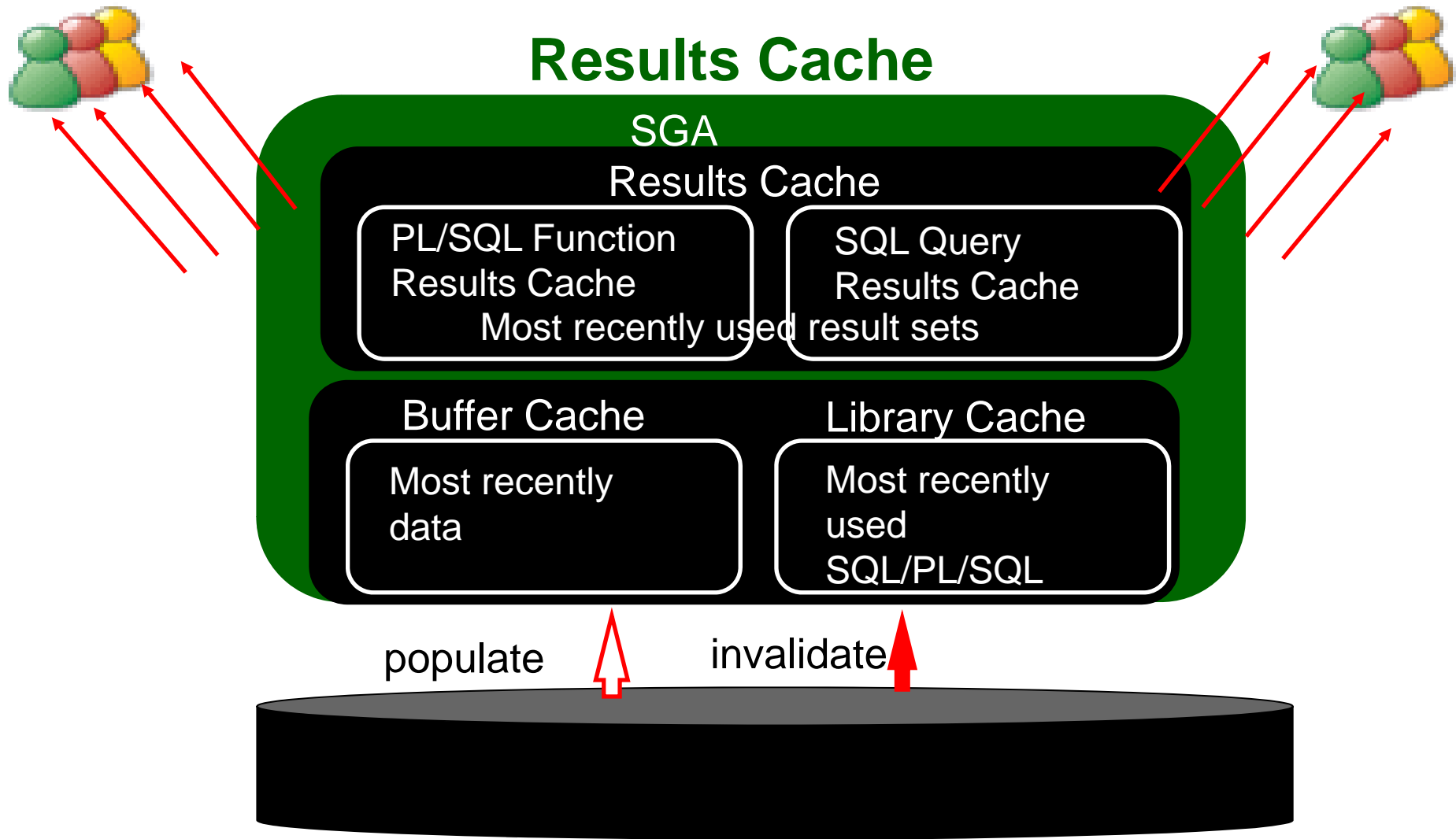
SQL\*Net message to client  
SQL\*Net message from client  
SQL\*Net break to client  
asynch description

OVERALL TOTALS FOR ALL NON-RECURSIVE STATEMENTS

call	count
Parse	262
Execute	371
Fetch	304
total	937

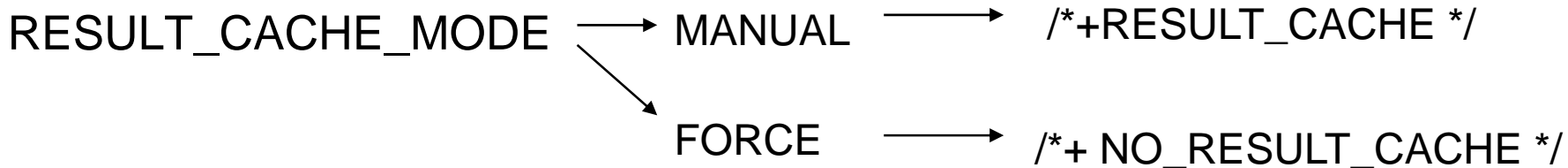
0.07	0.09	0	0	0	0
0.03	0.07	0	0	0	100
0.04	0.02	0	1836	0	224
0.15	0.20	0	1836	0	324

Misses in library cache during parse: 115





# Query Results Cache



```
SELECT /*+ RESULT_CACHE */  
      count(b.comments)  
FROM train1.events_large e, train1.bookings_large b  
WHERE e.org_id = :v1  
AND   e.event_no = b.event_no  
AND   e.comments = :v2;
```

# 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
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;

    RETURN (v_total_booked);
END;
```

# Monitoring the Results Cache

SELECT \* FROM v\$result\_cache\_memory

SELECT \* FROM v\$result\_cache\_objects

SELECT \* FROM v\$result\_cache\_statistics

SE

RI

# DEMO

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



# 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



# Gathering Statistics

Early CBO: “Make sure you gather statistics regularly”

Later CBO: “Don’t gather statistics unless data patterns change”

BUT

If you have new majority values you need to recreate the histogram

# Gathering and Publishing

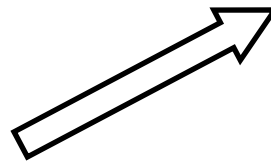
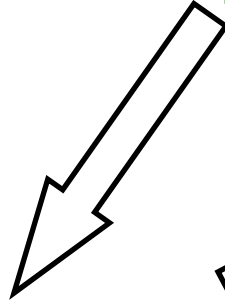
Set preferences to PUBLISH = 'FALSE'



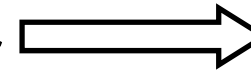
Gather Statistics

user\_tab\_pending\_stats  
user\_ind\_pending\_stats  
user\_col\_pending\_stats

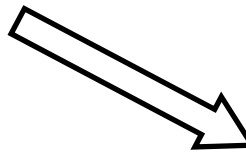
user\_tab\_statistics  
user\_ind\_statistics  
user\_tab\_col\_statistics



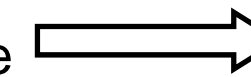
Better



dbms\_stats.  
publish\_pending\_stats



Worse



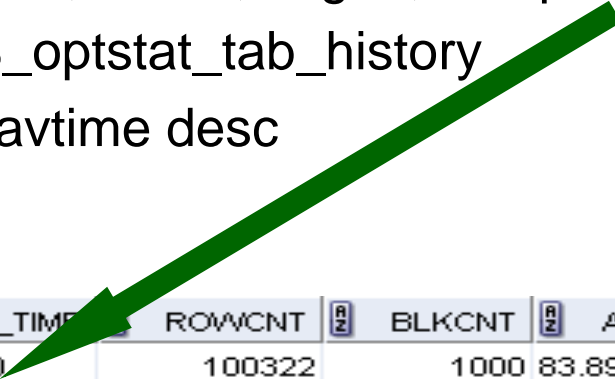
dbms\_stats.  
delete\_pending\_stats

Run test case

Alter session set  
optimizer\_pending\_statistics  
= TRUE

# Gathering and Publishing

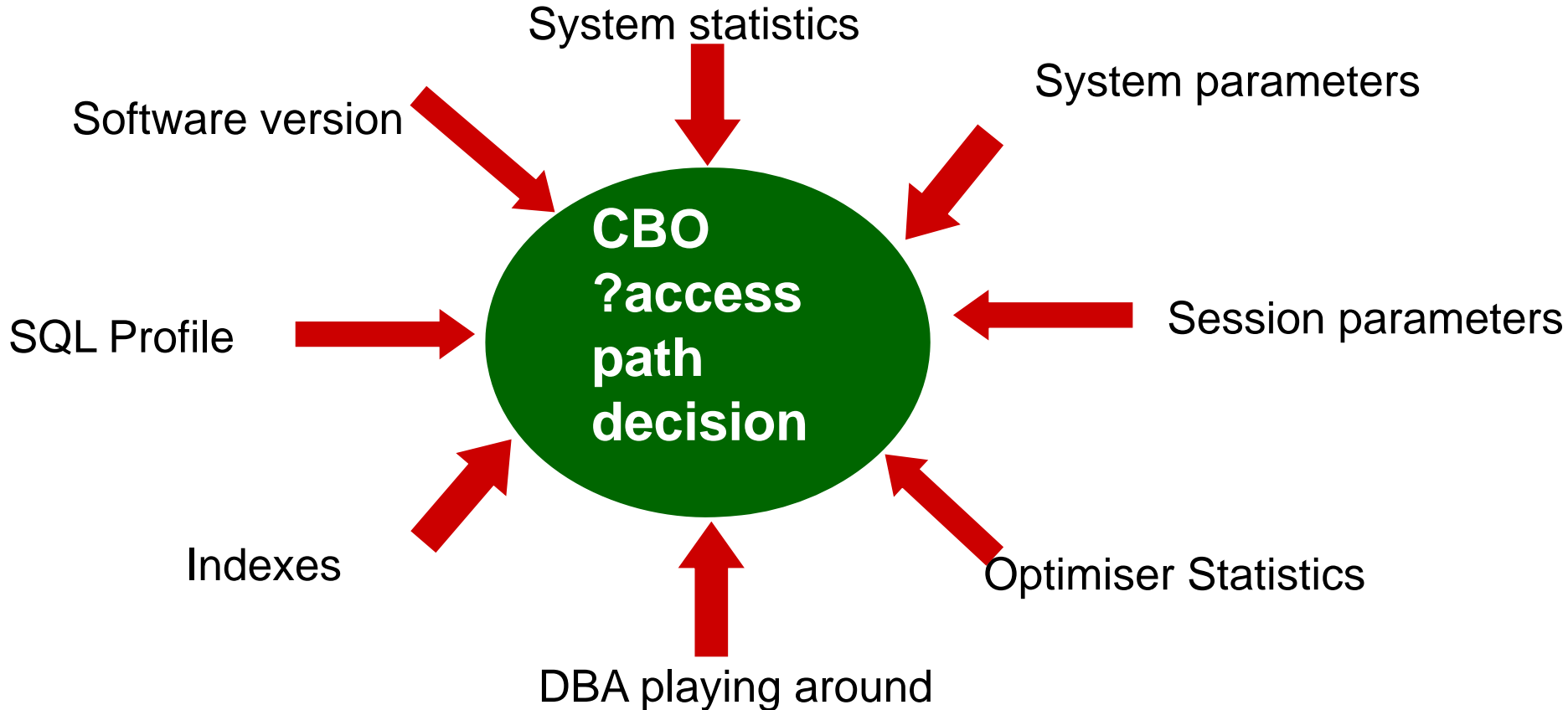
```
SELECT obj#, TO_CHAR(savtime,'dd/mm/yyyy') save_time,  
       rowcnt, blkcnt, avgrln, samplesize, analyzetime  
FROM   wri$_optstat_tab_history  
ORDER BY savtime desc
```



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



# 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 Management Base

Baseline =  
(Stored and Accepted plans)

Auto accept of new plan  
(if it performs better)

DBMS\_SPM.EVOLVE\_SQL\_PLAN\_BASELINE

Stored not accepted





# **SAGE Computing Services**

Customised Oracle Training Workshops and Consulting

## **Questions?**

*[www.sagecomputing.com.au](http://www.sagecomputing.com.au)*

*[penny@sagecomputing.com.au](mailto:penny@sagecomputing.com.au)*