



# **SAGE Computing Services**

Customised Oracle Training Workshops and Consulting

## **Bind Peeking – The Endless Tuning Nightmare**

***Penny Cookson***

***Managing Director and Principal Consultant***

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

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

** Oracle ACE**

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

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



# Characteristics

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

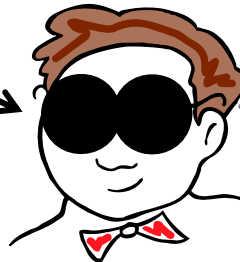
Thank you so much you've fixed it (I haven't done anything)

# Oracle Version < 9

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



'BRLG'



How many rows do I expect?



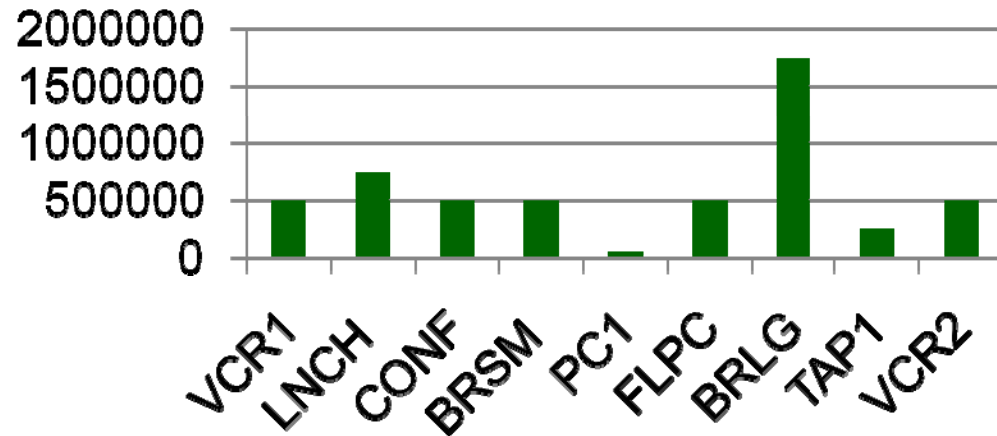
'PC1'

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;
```



'PC1'



How many rows do I expect?



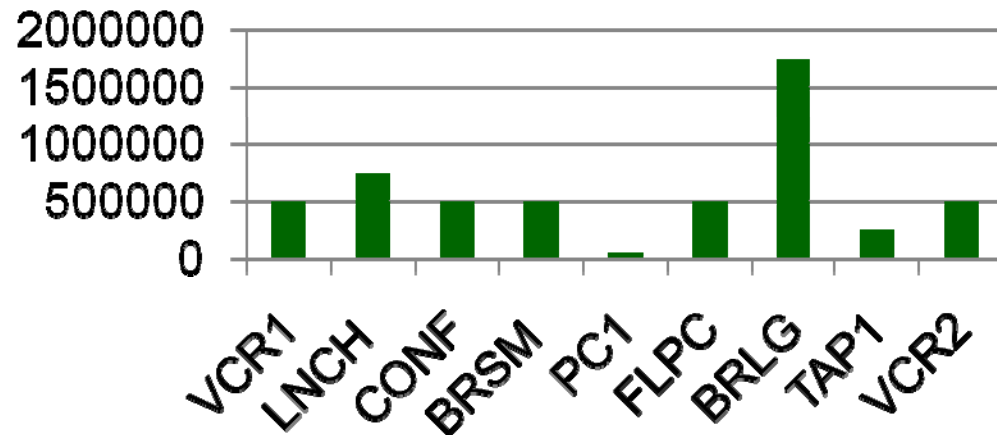
'BRLG'

Shared Pool

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

FULL SCAN

<Version 9 database  
No bind peeking



# What is Bind Peeking?

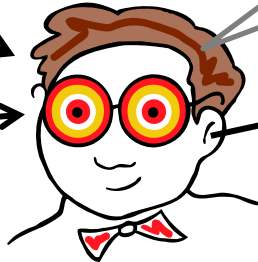
```
SELECT COUNT(l.quantity) FROM
bookings_skew l WHERE
resource_code = :v1;
```



'BRLG'



'PC1'



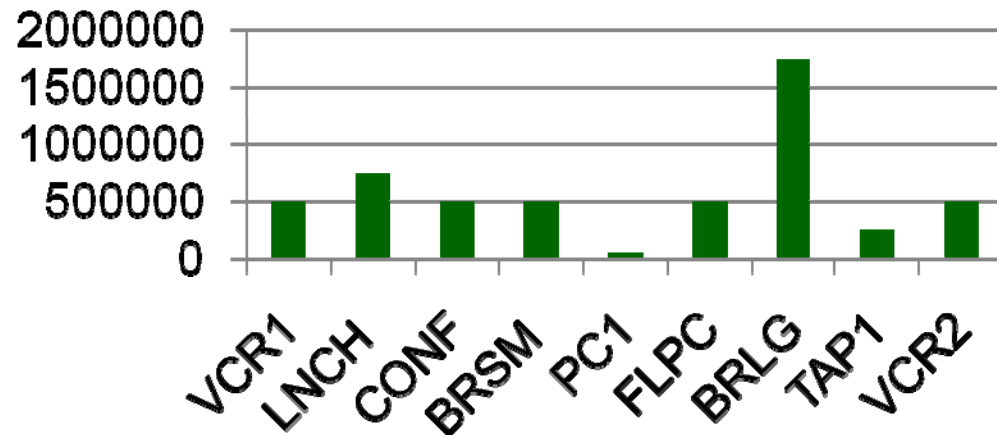
How many rows do I expect?

Shared Pool

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

FULL SCAN

>=Version 9 database  
Bind peeking



# What is Bind Peeking?

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

How many rows do I expect?

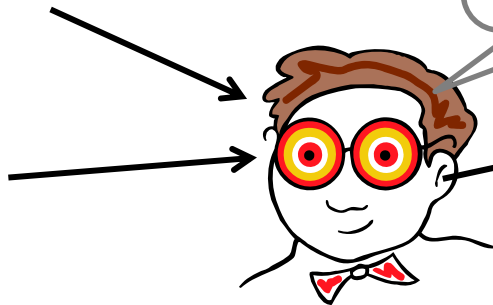
Shared Pool

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

INDEXED ACCESS



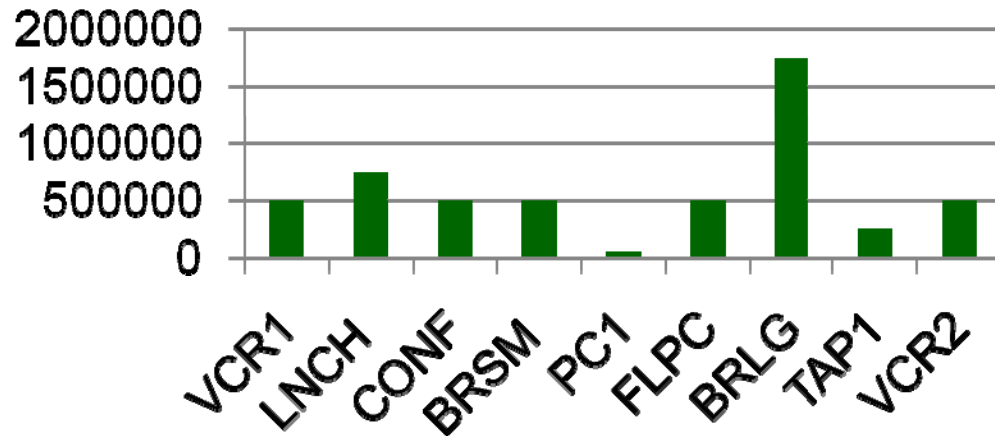
'PC1'



'BRLG'



>=Version 9 database  
Bind peeking





# No Histogram

\*\*\*\*\*

## SINGLE TABLE ACCESS PATH

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

AvgLen: 5.00 **NDV: 9** Nulls: 0 Density: 0.11111

Table: BOOKINGS\_SKEW Alias: L

Card: **Original: 5527238** **Rounded: 614138** Computed: 614137.56 Non

Adjusted: 614137.56

Access Path: TableScan

Cost: 7701.80 Resp: 7701.80 Degree: 0

Cost\_io: 7426.00 Cost\_cpu: 1605072831

Resp\_io: 7426.00 Resp\_cpu: 1605072831

Access Path: index (AllEqRange)

Index: BK\_RESSKEW

resc\_io: 32580.00 resc\_cpu: 509053165

ix\_sel: 0.11111 ix\_sel\_with\_filters: 0.11111

Cost: 32667.47 Resp: 32667.47 Degree: 1

Best:: AccessPath: TableScan

Cost: 7701.80 Degree: 1 Resp: 7701.80 Card: 614137.56 Bytes: 0

\*\*\*\*\*

**5527238/9 = 614138**



# Histogram – Majority 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: 1727558** Computed: 1727558.30 Non  
Adjusted: 1727558.30

Access Path: TableScan

Cost: 7710.70 Resp: 7710.70 Degree: 0

Cost\_io: 7426.00 Cost\_cpu: 1656892911

Resp\_io: 7426.00 Resp\_cpu: 1656892911

Access Path: index (AllEqRange)

Index: BK\_RESSKEW

resc\_io: 94139.00 resc\_cpu: 1468544140

ix\_sel: 0.31612 ix\_sel\_with\_filters: 0.31612

Cost: 94391.34 Resp: 94391.34 Degree: 1

Best:: AccessPath: TableScan

Cost: 7710.70 Degree: 1 Resp: 7710.70 Card: 1727558.30 Bytes: 0

\*\*\*\*\*

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





# 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 OK at

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	



# Histograms

Versions < 9: Histograms are no use with bind variables

Versions >= 9: Histograms are worse than no use  
with bind variables

Unless...

Each distinct SQL uses only either minority or majority  
values but not both

Or you don't keep/use the statements in the shared pool  
(in which case you might as well use literals)



# So if I have no Skewed Data I am OK?

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	



# Minority Range First

\*\*\*\*\*

## SINGLE TABLE ACCESS PATH

Column (#2): EVENT\_NO(NUMBER)

AvgLen: 5.00 NDV: 99178 Nulls: 0 Density: 1.0083e-005 Min: 211 Max: 100175

Table: BOOKINGS\_EVEN Alias: L

Card: Original: 5522697 Rounded: 4973 Computed: 4972.66 Non Adjusted: 4972.66

Access Path: TableScan

Cost: 7742.79 Resp: 7742.79 Degree: 0

Cost\_io: 7492.00 Cost\_cpu: 1459499283

Resp\_io: 7492.00 Resp\_cpu: 1459499283

Access Path: index (RangeScan)

Index: BK\_EVTEVEN

resc\_io: 4992.00 resc\_cpu: 37790328

ix\_sel: 9.0040e-004 ix\_sel\_with\_filters: 9.0040e-004

Cost: 4998.49 Resp: 4998.49 Degree: 1

Best:: AccessPath: IndexRange Index: BK\_EVTEVEN

Cost: 4998.49 Degree: 1 Resp: 4998.49 Card: 4972.66 Bytes: 0

\*\*\*\*\*

MIN	MAX
100	100200



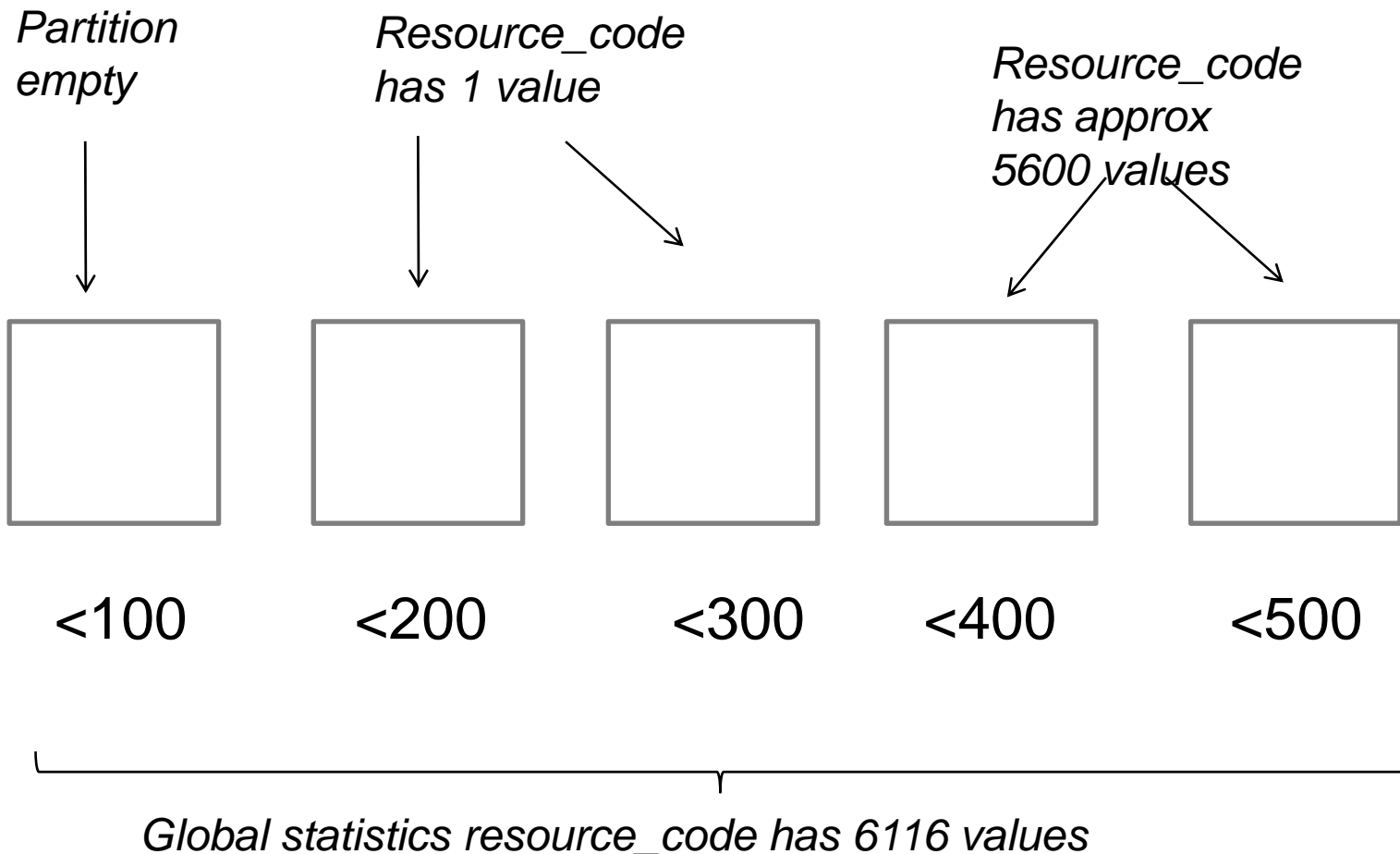
# So if I have no Skewed Data I am OK?

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	



# Partitions?

*Which statistics shall we use?*





# What Can We do?

## Own code:-

- Use Literals – but only for skewed data

- Write separate code for minority/majority

  - Give user separate menu options

  - Daily report v Annual report

  - Build sql dynamically with hints for various cases

  - User not aware

- (and keep your histograms)





## What Can We do?

Package:-

If we have no histogram what is it doing? → Full scan

Is a full scan acceptable? Yes

No histogram

Turn off bind peeking

`_optim_peek_user_binds=false`

Is a full scan acceptable? No

Run the majority ones first

Different session for majority/minority cases and change session variables



# What Can We do?

## Package:-

If we have no histogram what is it doing? → Index scan

(So none of these will help

- No histogram

- Turn off bind peeking

- `_optim_peek_user_binds=false)`

Remove index?

- Create an outline and force a full scan

- Run the majority ones first

Different session for majority/minority cases and change session variables





# Oracle 11g - Data

## BOOKINGS\_EVEN

RESO	NUM
-----	-----
VCR1	524288
LNCH	786432
CONF	524288
BRSM	786432
PC1	262144
FLPC	524288
BRLG	1310720
TAP1	262144
VCR2	524288
	-----
sum	5505024



# Oracle 11g - Data

## BOOKINGS\_SKEW

RESO	NUM	RESO	NUM
PC2	99	PC1	47865
PC3	99	TAP1	247853
PC4	99	FLPC	495713
PC5	99	CONF	495714
PC6	99	VCR2	495718
PC7	99	VCR1	495720
PC8	99	BRSM	743571
PC9	99	LNCH	743581
PC11	101	BRLG	1737243
PC12	102		
PC13	103		
PC14	104	sum	5505024
PC15	199		
PC16	399		
PC17	799		
PC18	999		
PC19	1999		
PC20	3999		
PC10	199		



## Bind Peeking + Adaptive Cursors Summary

Statements with bind variables (+histograms) 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

New values will use a plan for the appropriate selectivity range

Be careful when statements become invalidated by:-

- Gathering statistics

- Flushing the shared pool

- Restarting the database

- Or when they are aged out



# Adaptive Cursors Functionality

## Adaptive Cursors 11.1.0.6

Bind variable with Equality and Histogram

Not for range conditions

## Adaptive Cursors 11.1.0.7

Bind variable with Equality and Histogram

Range conditions

Do Not Support LIKE

## Future?

LIKE

SELECT /\*+ HINT TO MAKE IT BIND AWARE \*/



# Gathering Statistics

Early CBO: “Make sure you gather statistics regularly”

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

BUT

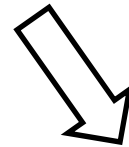
If you have new majority values you need to recreate the histogram



# 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

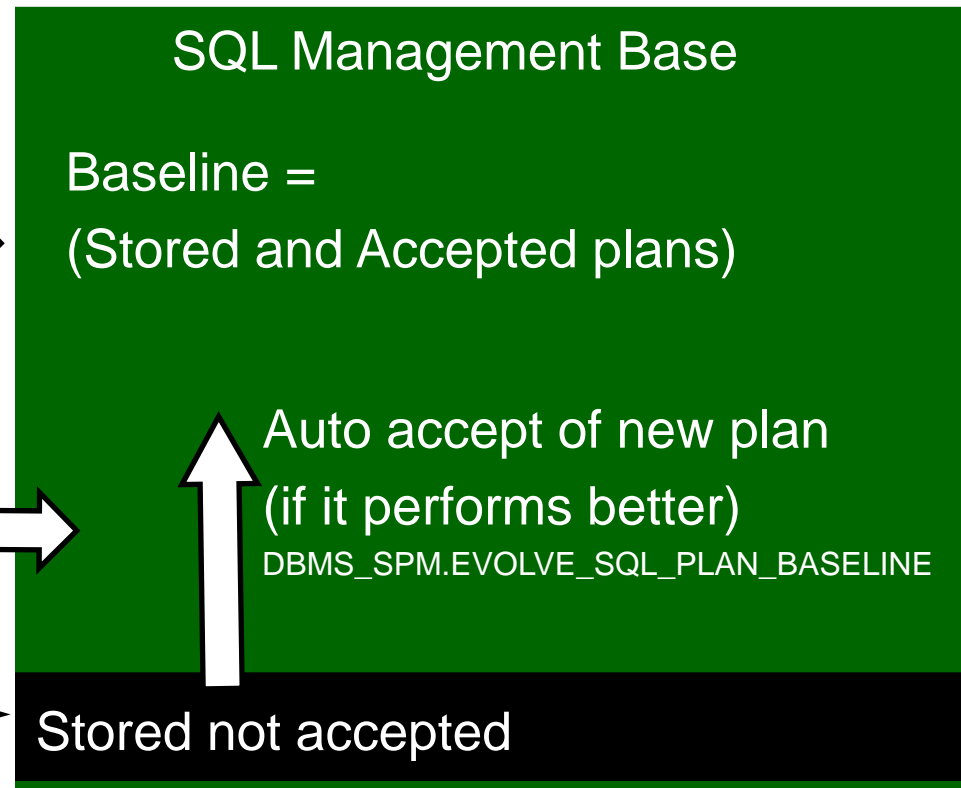
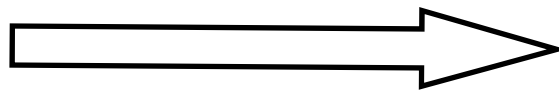
DBMS\_SPM.LOAD\_PLANS\_FROM\_SQLSET  
DBMS\_SPM.LOAD\_PLANS\_FROM\_CURSOR\_CACHE



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

# 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

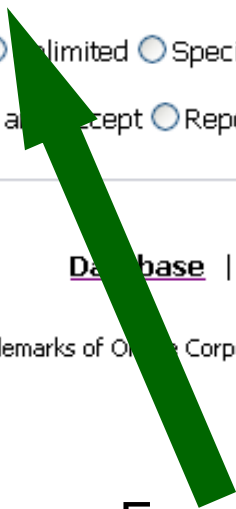
Plans that have not yet been accepted can be evolved (verified) to confirm they are suitable plan baselines.

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

Verify Performance  Yes  No

Time Limit  Auto  Unlimited  Specify  (minutes)

Action  Report and accept  Report only



Force Acceptance of the plan



# **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)*