

Upgrading to Oracle 11g



whoami

- **Kerry Osborne**
 - Senior Oracle Guy
 - Worked with V2-11g
 - My Oracle Blog: kerryosborne.oracle-guy.com



An IT Services Firm that Only does Oracle

- the obligatory marketing slide

- **Enkitech Basics**
 - Oracle-centric Consulting Services
 - Deeply skilled workforce / average of 15 years experience
 - Broad Coverage of Oracle Technology Products
- **Enkitech Lines of Business**
 - Oracle On-Call Services (remote DBA)
 - Oracle Database Technologies
 - Oracle Development Solutions
 - Oracle Security and Identity/Access Management
 - Oracle Business Intelligence

Oracle 11g

- **Initial Release (11.1.0.6) – July 2007**
- **Point Release (11.1.0.7) – September 2008**
- **R2 (11.2.0.1) – September 2009**
- **18% adoption of 11gR1** (Forester estimate, 9/09)
 - **Oracle has trained us that R1 is not a smart move**
 - **I believe we're witnessing a changing of the guard**

Why Now

- **R2 traditionally signals mass movement**
 - You want to stay in the middle of the herd



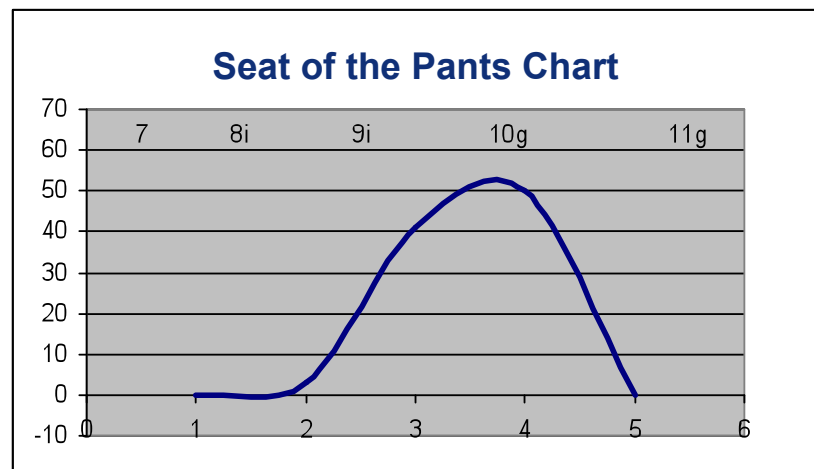
- **10g due to be de-supported next summer**
 - Extended Support fees waived until 2011

Flash Back to a couple of years ago

Top 10 reasons why you
should upgrade to 10g ...

1. You'll get fired if you don't

Well maybe not, but your boss won't be happy if the company starts having to pay extended support fees next summer.



Back to the Topic at Hand



So What Makes Things Go Wonky

(that's a technical term)

- **Changes in the Optimizer Behavior**
- **Changes in Default Values of Parameters**
- **Changes in Statistics Generation**

Digression – 9i to 10g

- **Biggest Issue Was Change in Stats**
 - 10g added a scheduled job to gather stats
 - Default in 9i was no Histograms & 100% sample size
 - Default in 10g was Histograms & auto_sample_size
 - **Result - Greatly Increased Plan Instability**
 - due primarily to Bind Variable Peeking
 - and inaccurate stats (NDV and histograms)
 - small sample sizes didn't help

“Those who cannot remember the past are doomed to repeat it.”

Oracle 11g – Things you should know

- **New Automatic Stats Gathering Job**
 - Behaves very similarly to 10g GATHER_STATS_JOB
 - DBMS_STATS.GATHER_DATABASE_STATS_JOB_PROC
 - New scheduling thingy called Autotask
 - Prioritizes and runs until Window expires
- **New Sampling Algorithm**
 - Fast NDV – about the same time as 10% sample size
 - Much much much better accuracy

Oracle 11g – Things you should know

New Thing Called an Autotask

```
SQL> select task_name,operation_name,status
2   from dba_autotask_task
3   /
```

TASK_NAME	OPERATION_NAME	STATUS
auto_space_advisor_prog	auto space advisor job	ENABLED
gather_stats_prog	auto optimizer stats job	ENABLED
AUTO_SQL_TUNING_PROG	automatic sql tuning task	ENABLED

- Use DBMS_AUTO_TASK_ADMIN to modify
- Whole set of views like DBA_AUTOTASK_XXXX

Oracle 11g – Things you should know

- **How to Disable Automatic Stats Gathering Job**

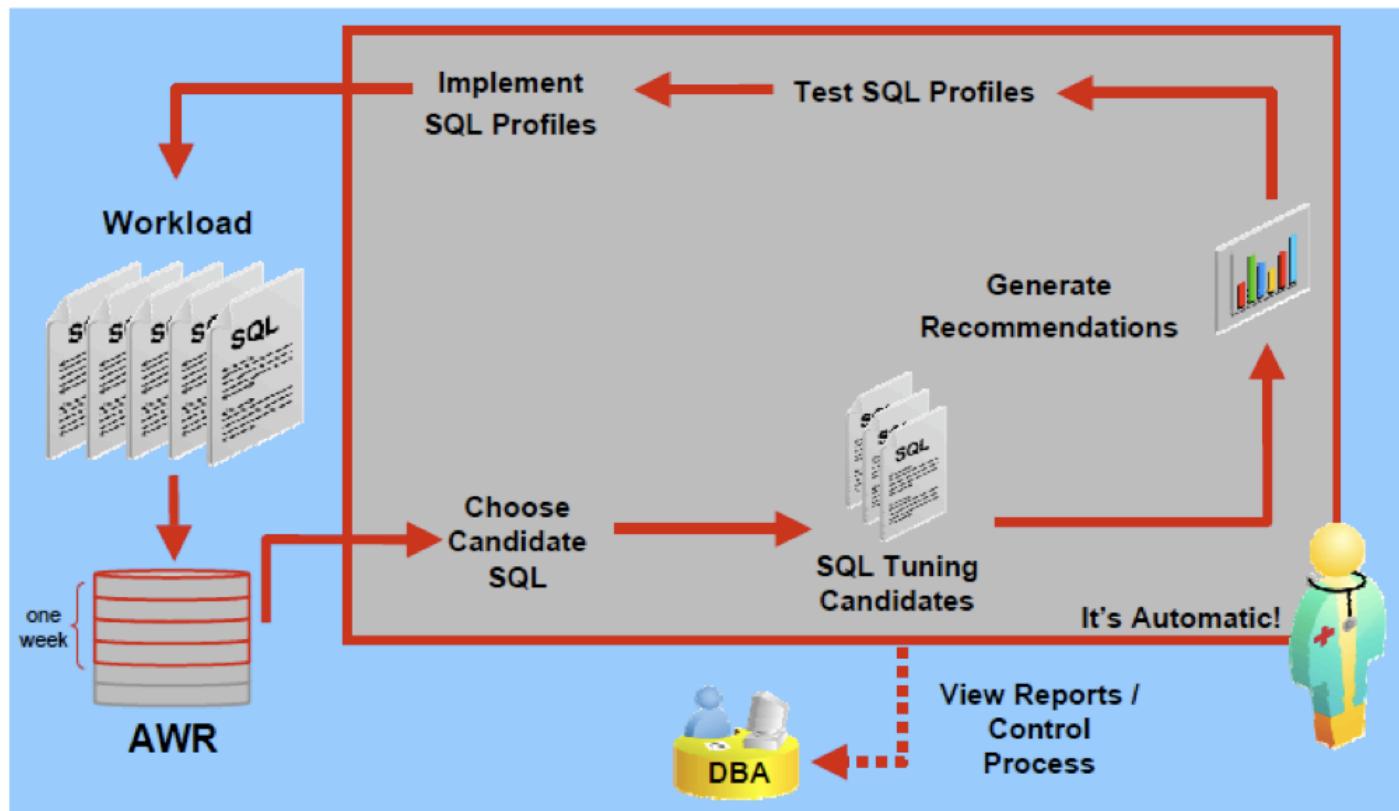
```
DBMS_AUTO_TASK_ADMIN.DISABLE (  
  client_name => 'auto optimizer stats collection',  
  operation => NULL,  
  window_name => NULL);
```

- **Consider setting AUTOSTATS_TARGET=ORACLE**

```
DBMS_STATS.SET_GLOBAL_PREFS ('AUTOSTATS' , 'ORACLE');
```

Automatic SQL Tuning Task

The Self-Managing Database



Oracle 11g – Things you should know

- **Predefined Windows**

```
SQL> select * from DBA_AUTOTASK_WINDOW_CLIENTS;
```

WINDOW_NAME	WINDOW_NEXT_TIME	WINDO	AUTOTASK	STATS	SEGMENT_ADV	SQL_TUNE_ADV	HEALTH_MONITOR
MONDAY_WINDOW	27-OCT-08 10.00.00.000000 PM CST6CDT	FALSE	ENABLED	ENABLED	ENABLED	ENABLED	DISABLED
TUESDAY_WINDOW	28-OCT-08 10.00.00.000000 PM CST6CDT	FALSE	ENABLED	ENABLED	ENABLED	ENABLED	DISABLED
WEDNESDAY_WINDOW	22-OCT-08 10.00.00.000000 PM CST6CDT	FALSE	ENABLED	ENABLED	ENABLED	ENABLED	DISABLED
THURSDAY_WINDOW	23-OCT-08 10.00.00.000000 PM CST6CDT	FALSE	ENABLED	ENABLED	ENABLED	ENABLED	DISABLED
FRIDAY_WINDOW	24-OCT-08 10.00.00.000000 PM CST6CDT	FALSE	ENABLED	ENABLED	ENABLED	ENABLED	DISABLED
SATURDAY_WINDOW	25-OCT-08 06.00.00.000000 AM CST6CDT	FALSE	ENABLED	ENABLED	ENABLED	ENABLED	DISABLED
SUNDAY_WINDOW	26-OCT-08 06.00.00.000000 AM CST6CDT	FALSE	ENABLED	ENABLED	ENABLED	ENABLED	DISABLED

7 rows selected.

***11g Default is 10pm-2am weekdays and 6am-2am weekends**

***10g Default was 10pm-6am weekdays and all weekend**

Oracle 11g – Things you should know

- **Changed Defaults**

```
KSO@LAB112> @changed_defaults
```

NAME	VALUE_10	VALUE_11
-----	-----	-----
_awr_flush_threshold_metrics	FALSE	TRUE
_enable_NUMA_optimization	TRUE	FALSE
_enable_row_shipping	FALSE	TRUE
_lm_rcvr_hang_kill	FALSE	TRUE
_notify_crs	FALSE	TRUE
_optimizer_enable_density_improvements	FALSE	TRUE
_rcfg_disable_verify	FALSE	TRUE
_rm_numa_sched_enable	FALSE	TRUE
_xsqlapi_use_olap_dml_for_rank	FALSE	TRUE

```
9 rows selected.
```

* Pay particular attention to optimizer* parameters

Digression – New Object Types

```
SQL> select distinct object_type  
      from dba_objects order by 1;
```

```
OBJECT_TYPE
```

```
-----
```

```
CLUSTER  
CONSUMER GROUP  
CONTEXT  
DIMENSION  
DIRECTORY  
EDITION  
EVALUATION CONTEXT  
FUNCTION  
INDEX  
INDEX PARTITION  
INDEXTYPE  
JAVA CLASS  
JAVA DATA  
JAVA RESOURCE  
JOB  
JOB CLASS  
LIBRARY  
LOB  
LOB PARTITION  
MATERIALIZED VIEW  
OPERATOR
```

```
PACKAGE  
PACKAGE BODY  
PROCEDURE  
PROGRAM  
QUEUE  
RESOURCE PLAN  
RULE  
RULE SET  
SCHEDULE  
SEQUENCE  
SYNONYM  
TABLE  
TABLE PARTITION  
TRIGGER  
TYPE  
TYPE BODY  
UNDEFINED  
VIEW  
WINDOW  
WINDOW GROUP  
XML SCHEMA
```

```
42 rows selected.
```



Digression – New Object Types

```
KSO@LAB112> @obj
Enter value for owner:
Enter value for name:
Enter value for type: UNDEFINED
```

OWNER	OBJECT_NAME	OBJECT_TYPE	STATUS	T
SYS	CURRENT_OPEN_WINDOW	UNDEFINED	VALID	N
SYS	DEFAULT_TIMEZONE	UNDEFINED	VALID	N
SYS	EMAIL_SENDER	UNDEFINED	VALID	N
SYS	EMAIL_SERVER	UNDEFINED	VALID	N
SYS	EVENT_EXPIRY_TIME	UNDEFINED	VALID	N
SYS	FILE_WATCHER_COUNT	UNDEFINED	VALID	N
SYS	LAST_OBSERVED_EVENT	UNDEFINED	VALID	N
SYS	LOG_HISTORY	UNDEFINED	VALID	N
SYS	MAX_JOB_SLAVE_PROCESSES	UNDEFINED	VALID	N

```
9 rows selected.
```

Oracle 11g – Things you should know

AMM – Automatic Memory Management

9i - pga_aggregate_target

10g - sga_target

11g - memory_target

- Note bug 7272646 – 3G limit on 11.1.0.7 on 64bit Linux

Observations:

Wants to grab lot's for PGA

Wants to grab lot's for shared_pool

Doesn't want to give it back

Don't forget the __ parameters

Don't forget the shared_pool is used for a lot of things now

See Tanel's Post:

<http://blog.tanelpoder.com/2007/08/21/oracle-11g-internals-part-1-automatic-memory-management/>

Oracle 11g – Things you should know

- **Connection Auditing**
 - on by default but purging not enabled
 - AUD\$ table is in the SYSTEM tablespace
 - DBMS_AUDIT_MGMT
 - CREATE_PURGE_JOB
 - SET_AUDIT_TRAIL_LOCATION

11g New Features / Optimizer Changes

Plan Stability (SPM)

Adaptive Cursor Sharing (ACS)

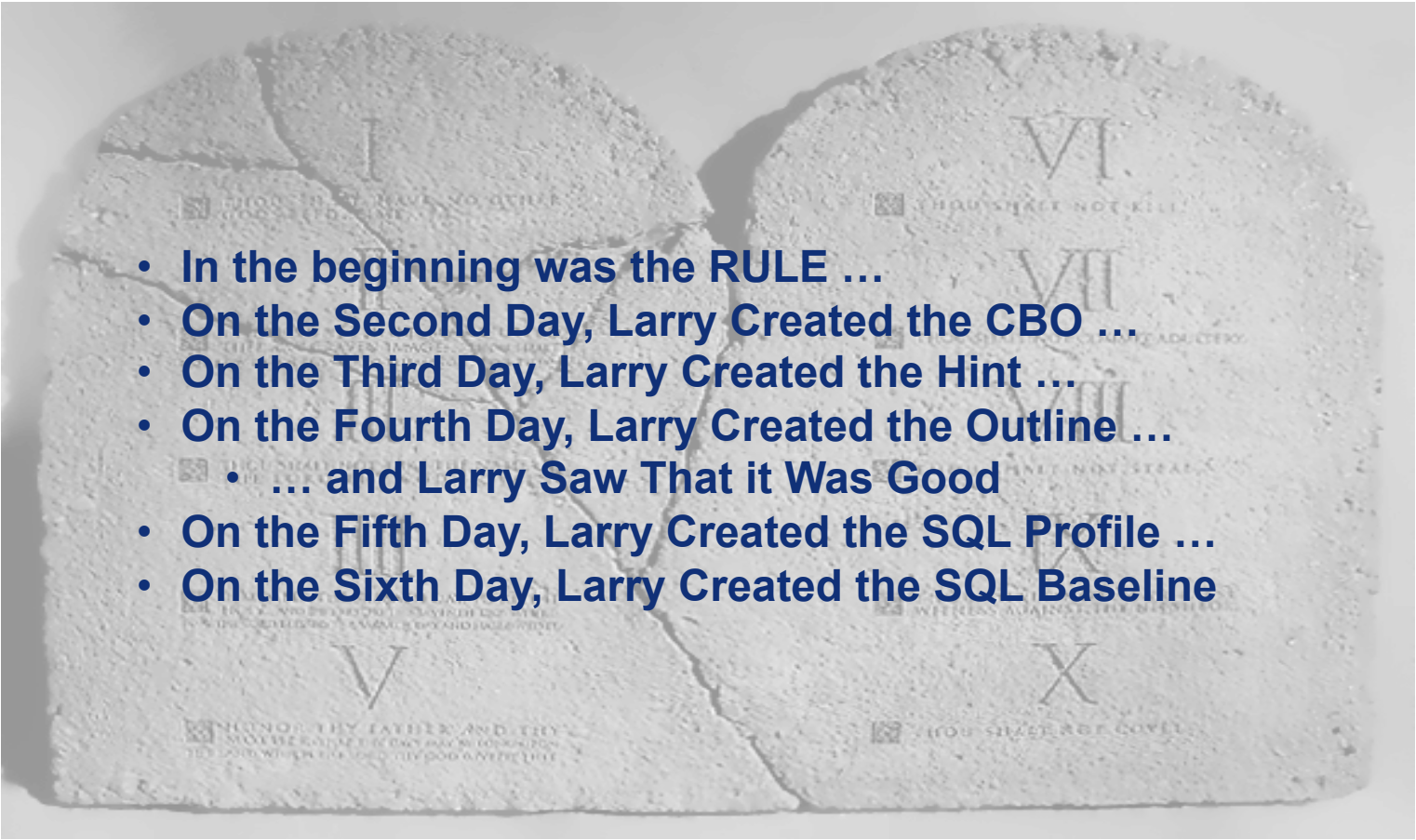
Invisible Indexes

Workload Capture and Replay (RAT)

Editions

Segment Creation on Demand

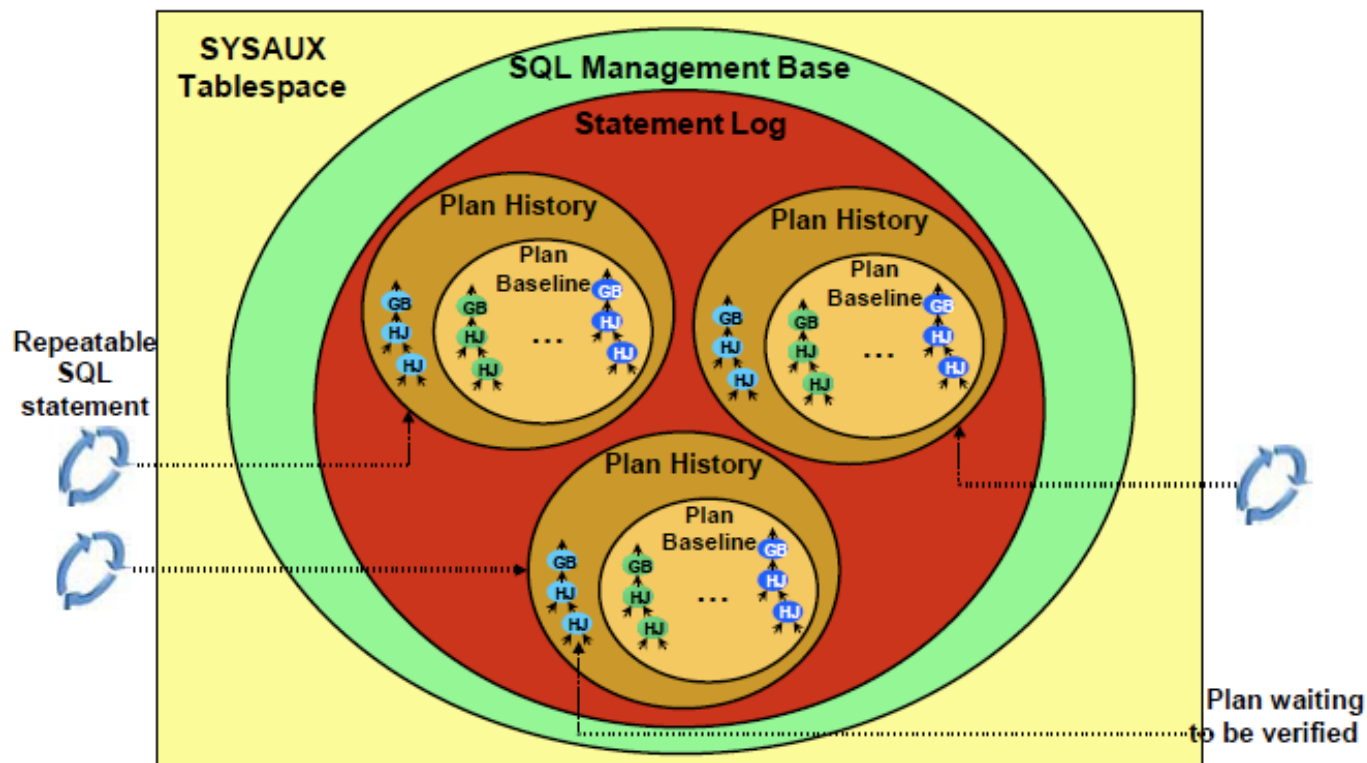
Plan Stability – A Brief History

- 
- A faded, grayscale image of the Ten Commandments stone tablets, serving as a background for the list. The tablets are arranged in two columns, with Roman numerals I through X visible on each. The text on the tablets is faint and mostly illegible due to the low contrast.
- **In the beginning was the RULE ...**
 - **On the Second Day, Larry Created the CBO ...**
 - **On the Third Day, Larry Created the Hint ...**
 - **On the Fourth Day, Larry Created the Outline ...**
 - **... and Larry Saw That it Was Good**
 - **On the Fifth Day, Larry Created the SQL Profile ...**
 - **On the Sixth Day, Larry Created the SQL Baseline**

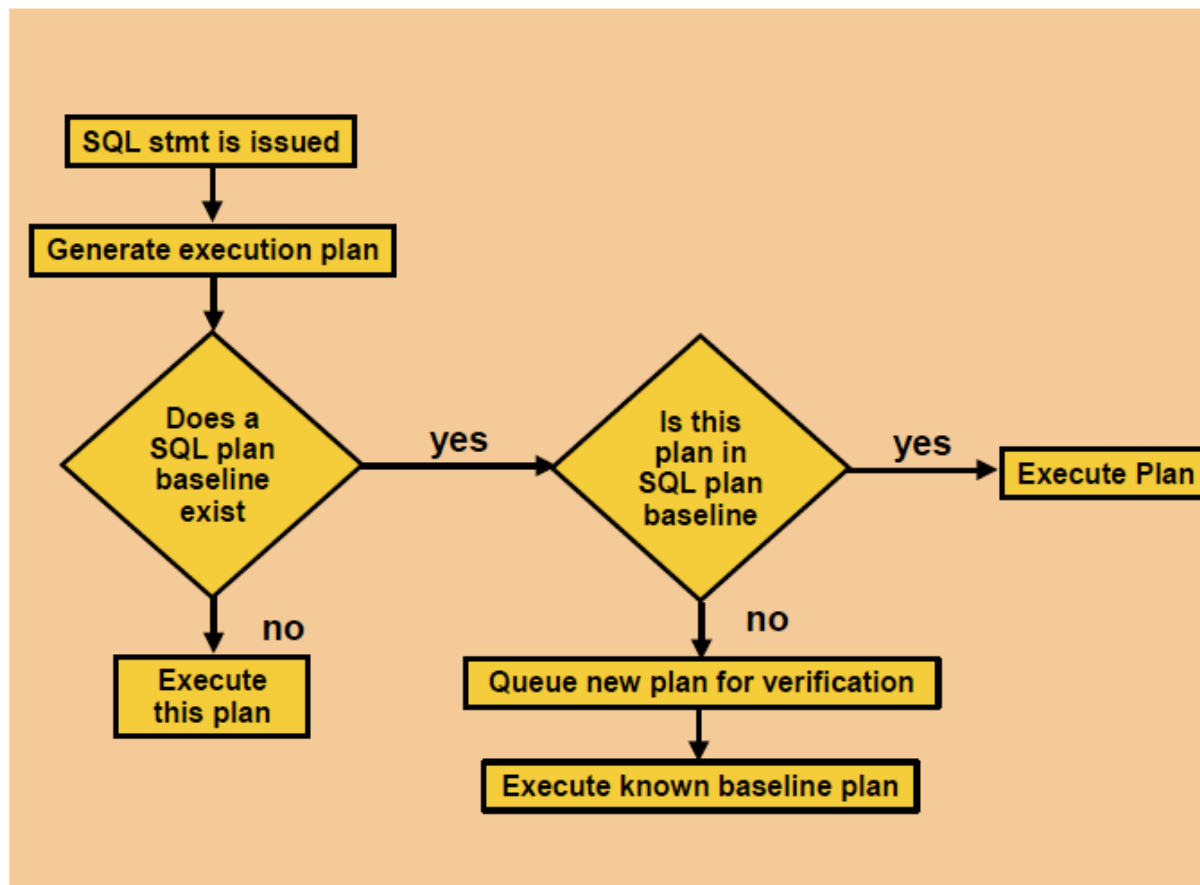
SQL Plan Management

- **The Idea is to Prevent Backward Movement**
- **New Framework using Baselines**
 - **SPM is Off by default (sort of)**
 - **optimizer_use_sql_plan_baselines=true**
 - **But no plans are Baselined by default**
 - **Baselines can be bulk loaded**
 - **From a SQL Tuning Set (10g)**
 - **From Outlines**
 - **From the cursor cache**
 - **Via optimizer_capture_sql_plan_baselines=true**

SQL Plan Management



SQL Plan Management – Hard Parse



SQL Plan Management – Hard Parse

On Hard Parse – Psuedo Code

```
IF statement not found in SMB THEN  
  parse and execute  
ELSE /* statement found in SMB */  
  parse (yields new_plan)  
  IF new_plan in SMB THEN  
    IF fixed=YES and accepted=YES THEN  
      execute new_plan  
    ELSE IF other fixed=YES and accepted=YES plans exist THEN  
      cost all fixed plans and execute lowest cost fixed plan  
    ELSE  
      cost all non-fixed plans and execute lowest cost plan  
    END IF  
  ELSE /* new_plan not in SMB */  
    add plan to SMB (accepted=NO)  
    IF fixed=YES and accepted=YES plans exist THEN  
      execute lowest cost fixed=YES and accepted=YES plan  
    ELSE  
      execute lowest cost accepted=YES plan  
    END IF  
END IF
```

SQL Plan Management

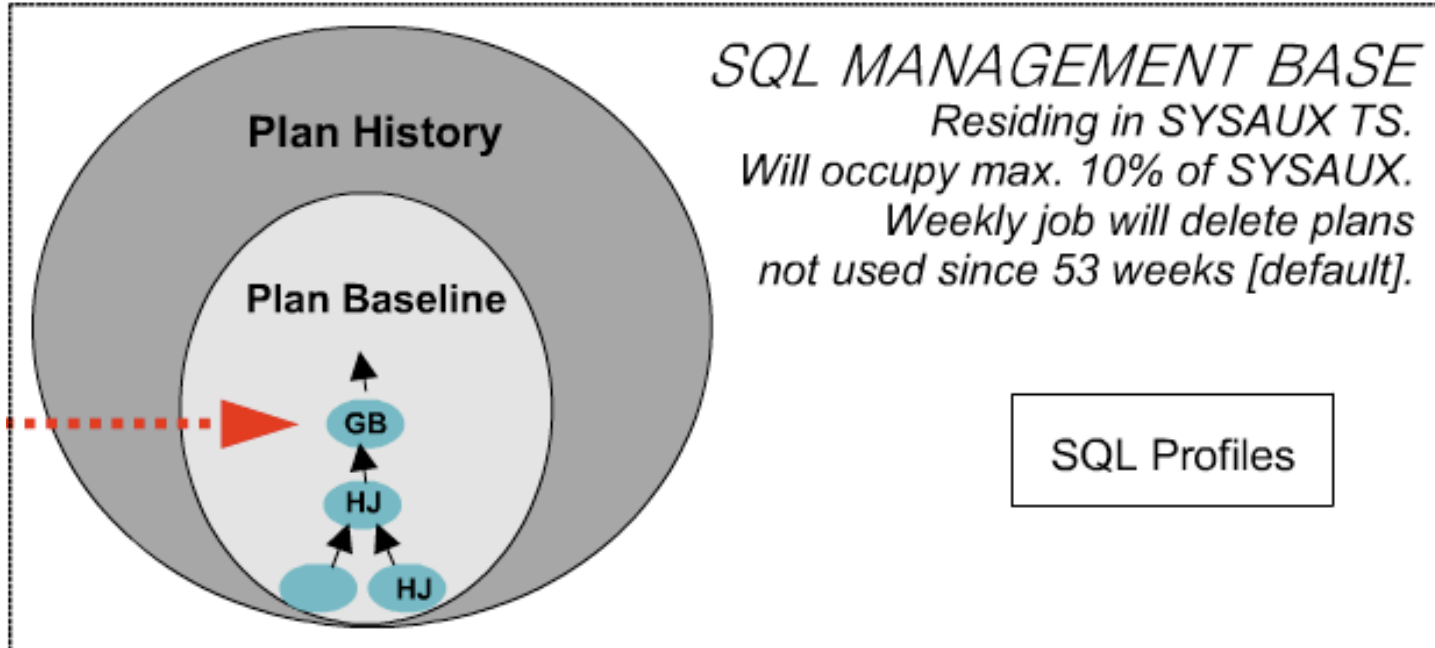
- So what's actually stored?
 - A plan hash value (calculated differently than v\$sql)
 - Hints to reproduce the plan
 - Signature (no sql_id)
 - The actual plan is not stored

```
SYS@LAB111> select spb.sql_handle, spb.plan_name, spb.sql_text,
2  spb.enabled, spb.accepted, spb.fixed,
3  to_char(spb.last_executed,'dd-mon-yy HH24:MI') last_executed
4  from
5  dba_sql_plan_baselines spb;
```

SQL_HANDLE	PLAN_NAME	SQL_TEXT	ENABLED	ACC	FIX	LAST_EXECUTED	
SYS_SQL_36bf1c88f777e894	SYS_SQL_PLAN_f777e89455381d08	select avg(pk_col)	f	YES	YES	NO	27-oct-09 10:20
SYS_SQL_f2784d83c1974f5e	SYS_SQL_PLAN_c1974f5e54680e33	select avg(pk_col)	f	YES	YES	NO	27-oct-09 11:12
SYS_SQL_f2784d83c1974f5e	SYS_SQL_PLAN_c1974f5e55381d08	select avg(pk_col)	f	YES	NO	NO	

...

SQL Plan Management



Change defaults with DBMS_SPM.CONFIGURE (50% of SYSAUX is max)

SQL Plan Management

Three very similar things –

- **Outlines**
- **SQL Profiles**
- **Baselines (fixed)**

- **Outline's deprecated ???**
 - **Docs still have it – but recommends against using them**
 - **Still possible to create and they do get used**
 - **The OUTLN schema has changed between 10gR2 and 11gR1**
 - **These changes indicate it hasn't been completely abandoned**
 - **Baselines or SQL Profiles are probably a better choice**

- **11gR2 has DBMS_SPM.MIGRATE_STORED_OUTLINE**
- **Also possible to create Baseline on statement using Outline**
- **Outlines take precedence – so you have to disable them**
- **SQL Profiles and Baselines can be used together (OPT_ESTIMATE)**

SQL Plan Management

Reminder - Baselines are Hint Based

```
SYS@LAB112> @baseline_hints
Enter value for baseline_plan_name: SQL_PLAN_3dgswj3vrgu4ned88b4f4
```

```
OUTLINE_HINTS
```

```
-----
INDEX(@"SEL$1" "A"@"SEL$1" ("SKEW"."COL2" "SKEW"."COL1"))
OUTLINE_LEAF(@"SEL$1")
ALL_ROWS
DB_VERSION('11.2.0.1')
OPTIMIZER_FEATURES_ENABLE('11.2.0.1')
IGNORE_OPTIM_EMBEDDED_HINTS
```

```
6 rows selected.
```

Digression – V\$SQL_HINT

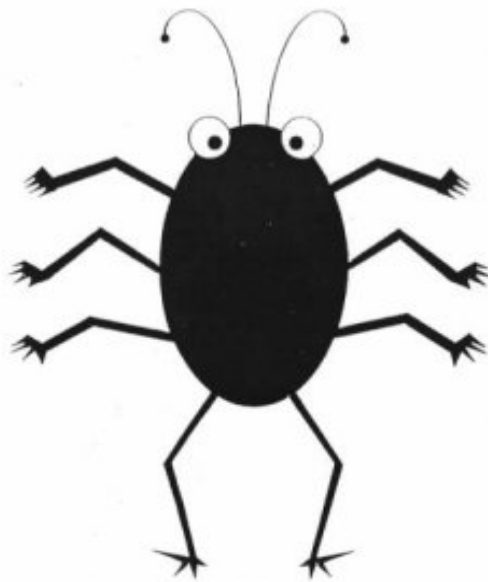
- Shows all hints and version they were introduced

```
SQL> @sql_hints
SQL> select name,version from v$sql_hint
  2  where upper(name) like '%'||upper(nvl('&hint',name))||'% '
  3  order by name;
Enter value for hint: INDEX%
```

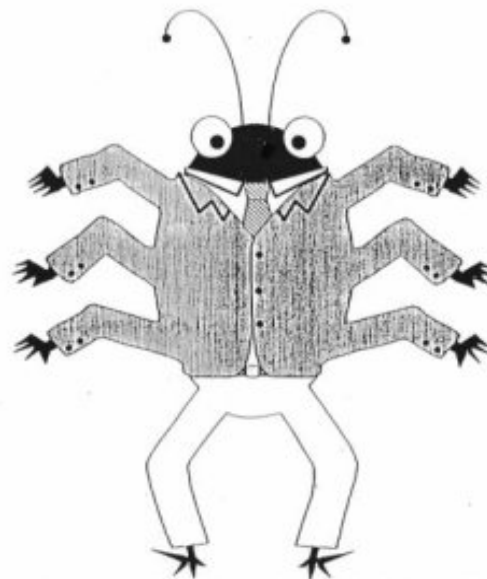
NAME	VERSION
INDEX	8.0.0
INDEX_ASC	8.1.0
INDEX_COMBINE	8.1.0
INDEX_DESC	8.1.0
INDEX_FFS	8.1.0
INDEX_JOIN	8.1.5
INDEX_RRS	9.0.0
INDEX_RS_ASC	11.1.0.6
INDEX_RS_DESC	11.1.0.6
INDEX_SS	9.0.0
INDEX_SS_ASC	9.0.0
INDEX_SS_DESC	9.0.0
INDEX_STATS	10.1.0.3

Digression – Bind Variable Peeking

Drives Me Nuts!



BUG



FEATURE

Adaptive Cursor Sharing

- **No more bind variable peeking issues – Yeah!!**
 - **One of the most pervasive performance issues with Oracle 10g**
 - **Allows multiple plans based on values of bind variables**
 - **Trade off between cost of parsing and developing optimal plans**
 - **ACS Means slightly longer parsing**
 - **OK because the optimizer can see the values of bind variables**
 - **ACS checks execution statistics to see if this is necessary**
 - **ACS will also merge cursors if plans are the same**

Adaptive Cursor Sharing – Bits and Pieces

A few new views –

- V\$SQL_CS_STATISTICS
- V\$SQL_CS_SELECTIVITY
- V\$SQL_CS_HISTOGRAM

A few new columns (in v\$sql and v\$sqlarea) –

- IS_BIND_SENSITIVE
- IS_BIND_AWARE
- IS_SHARABLE

```
SYS@LAB111> @find_sql_acs
Enter value for sql_text:
Enter value for sql_id: algwvb95akb9d
Enter value for is_bind_aware:
```

SQL_ID	CHILD	PLAN_HASH_VALUE	IBA	ISH	EXECS	AVG_ETIME	AVG_LIO	SQL_TEXT
algwvb95akb9d	0	3723858078	N	N	3	1.66	16,523	select avg(pk_col) f
algwvb95akb9d	1	568322376	Y	Y	2	6.67	162,297	select avg(pk_col) f
algwvb95akb9d	2	3723858078	Y	Y	6	.00	35	select avg(pk_col) f

Adaptive Cursor Sharing – Turning it Off

```
alter system set "_OPTIMIZER_EXTENDED_CURSOR_SHARING_REL"=none scope=spfile;  
alter system set "_OPTIMIZER_EXTENDED_CURSOR_SHARING"=none scope=spfile;  
alter system set "_OPTIMIZER_ADAPTIVE_CURSOR_SHARING"=false scope=spfile;
```

- **Then restart the instance**
- **Check Metalink for more details!**

*** Shouldn't need to do this in 11.1.0.7 and later**

Invisible Indexes

- **OPTIMIZER_USE_INVISIBLE_INDEXES=true**
- **Hide an Existing Index (instead of drop)**
- **Create a New Index for Testing**
- **Decouples Creation from Publication**

Workload Capture and Replay

(i.e. Real Application Testing)

- **DBMS_WORKLOAD_CAPTURE**
 - Back ported to 10.2.0.4 and 9.2.0.8
 - Allows include and exclude rules
 - Doesn't appear too intrusive (but be careful)
- **DBMS_WORKLOAD_REPLAY**
 - Only works on 11g
 - Starts worker processes
 - Replays transactions w/ accurate timing

*** Metalink note: 560977.1 – *Real Application Testing for Earlier Releases***

What Can We Do to Mitigate Risks

- 1. Keep a copy of optimizer stats**
- 2. Keep a copy of execution plans**
- 3. Keep historical performance statistics**
- 4. Need to test (duh!)**

Digression – my favorite basic approach

Hardware refresh and major database upgrade together!

- 1. Violates my “*Only Change One Thing at a Time*” rule**
- 2. Minimizes Impact to Business**
- 3. Refresh Schedules Often Coincide**
- 4. Presumably Better Hardware Will Help**
- 5. The Old System Will Still Be Available**
 - **If things go horribly wrong, you can fall back**
 - **You don’t have to be so careful about copying info**
 - **You can see exactly how things behaved before**

Copy Optimizer Stats

- **Easy!**
- **Export stats using dbms_stats package**

```
EXEC DBMS_STATS.CREATE_STATS_TABLE('SYS','MY_STATS_TAB');  
EXEC DBMS_STATS.EXPORT_SCHEMA_STATS('&schema_name','MY_STATS_TAB','10g_Stats');
```

Copy Performance Statistics

- **Not Too Hard!**
- **Most Info in AWR**
 - **Will still be there after upgrade**
- **Also in Statspack (level 7)**
 - **Export perfstat user if doing in-place upgrade**

*** Potential Problem – not all statements captured**

Copy Execution Plans

- **Not so easy *&%\$!**
- **AWR captures plans**
- **Statspack captures plans if level set to 6 or above**
- **Unfortunately, not all plans are captured**
- **Arguably the most important will be captured**
 - **But very efficient statements may not be**

Note: If you still have access to the old system all is well ...

Testing

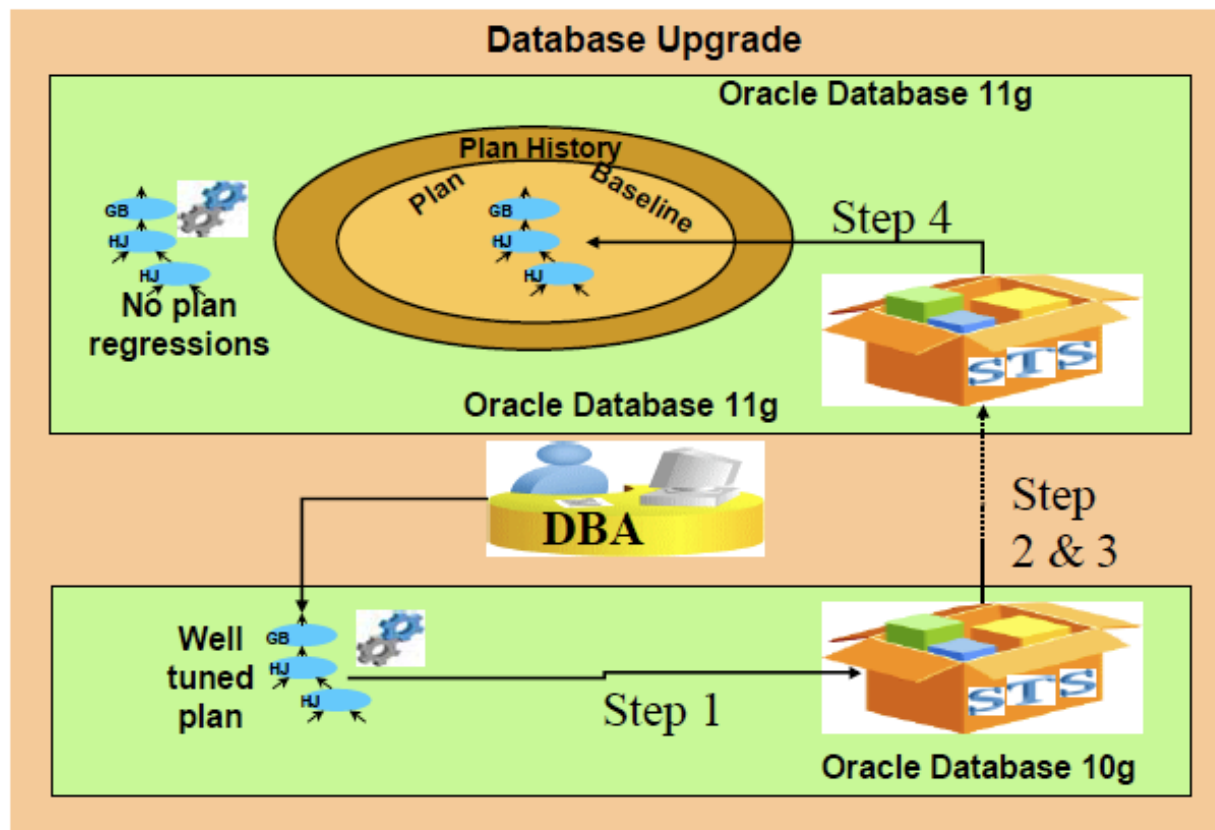
Please Allow Adequate Time for Testing!

- **Regression Test Suite – (Hammerora, Swingbench, ...)**
- **RAT can help if you don't have a good test suite**
- **RAT can also report on changed plans**
- **Suggestions**
 - **Try it without hints (optimizer_ignore_hints)**
 - **Try it without Outlines/SQL Profiles (ACS)**

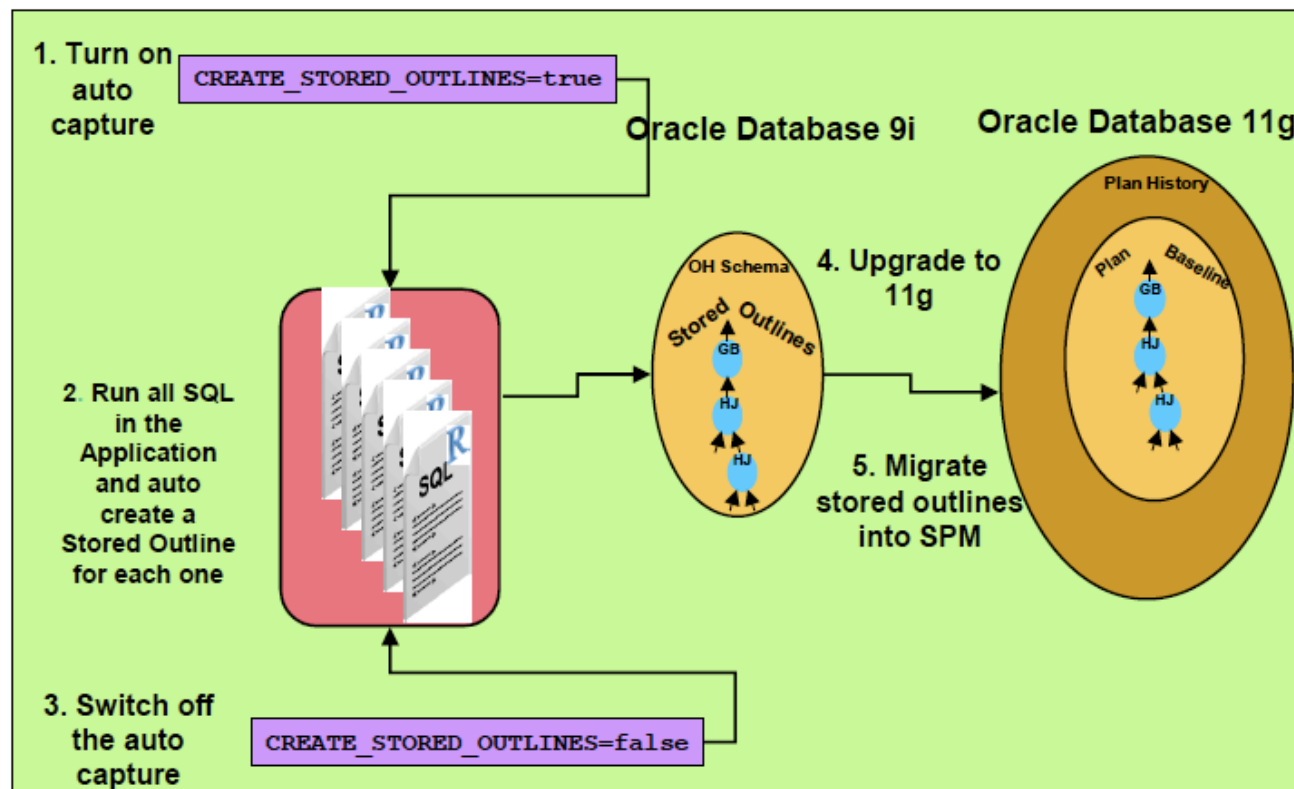
Testing – More Advanced Options

- **Use Old Plans to Seed SPM**
 - **SQL Tuning Sets (10gR2)**
 - **Outlines**
 - **Cursor Cache (set optimizer_features_enable)**
 - **SQL Trace**

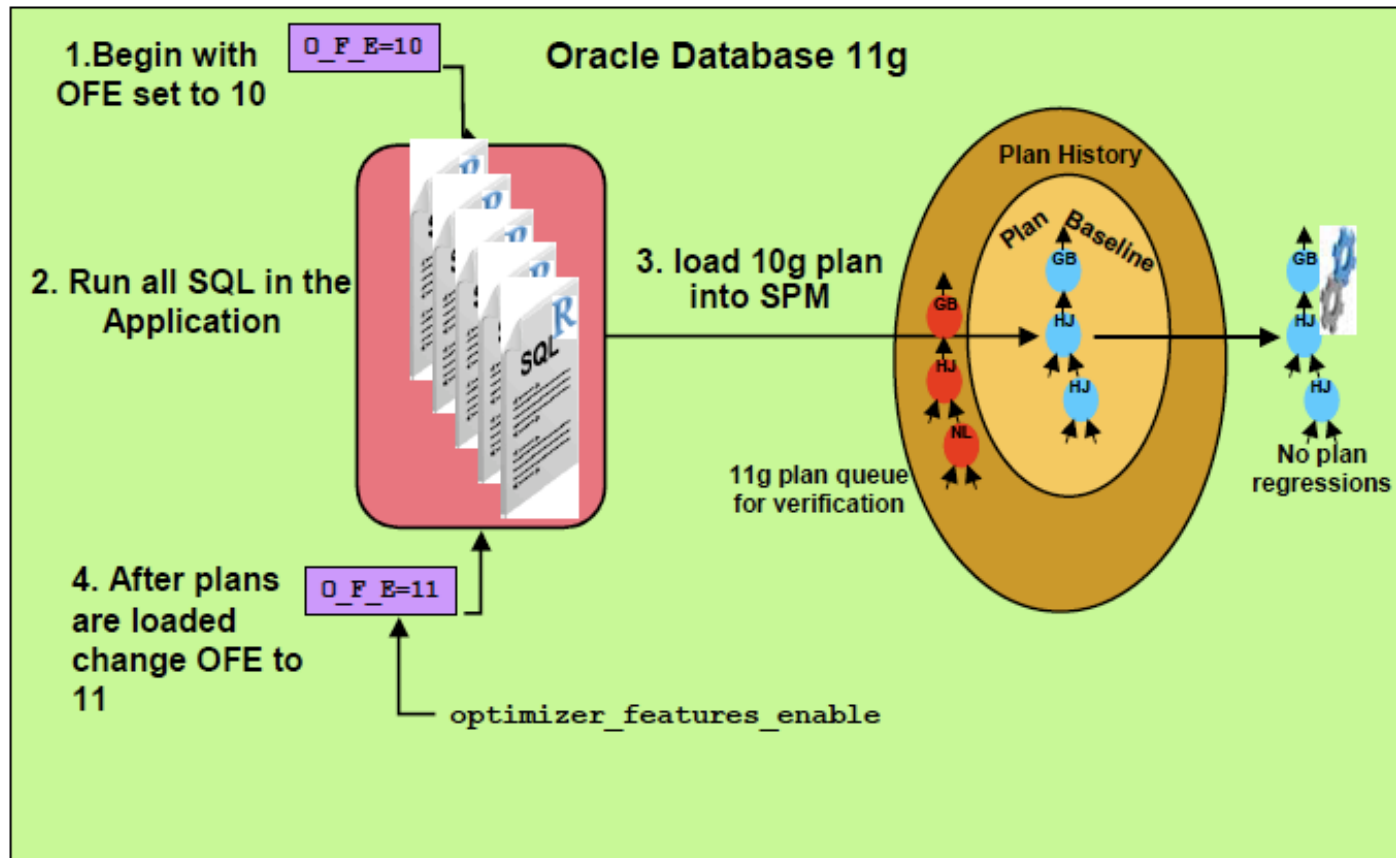
Seeding SPM – SQL Tuning Sets



Seeding SPM - Outlines



Seeding SPM – Cursor Cache

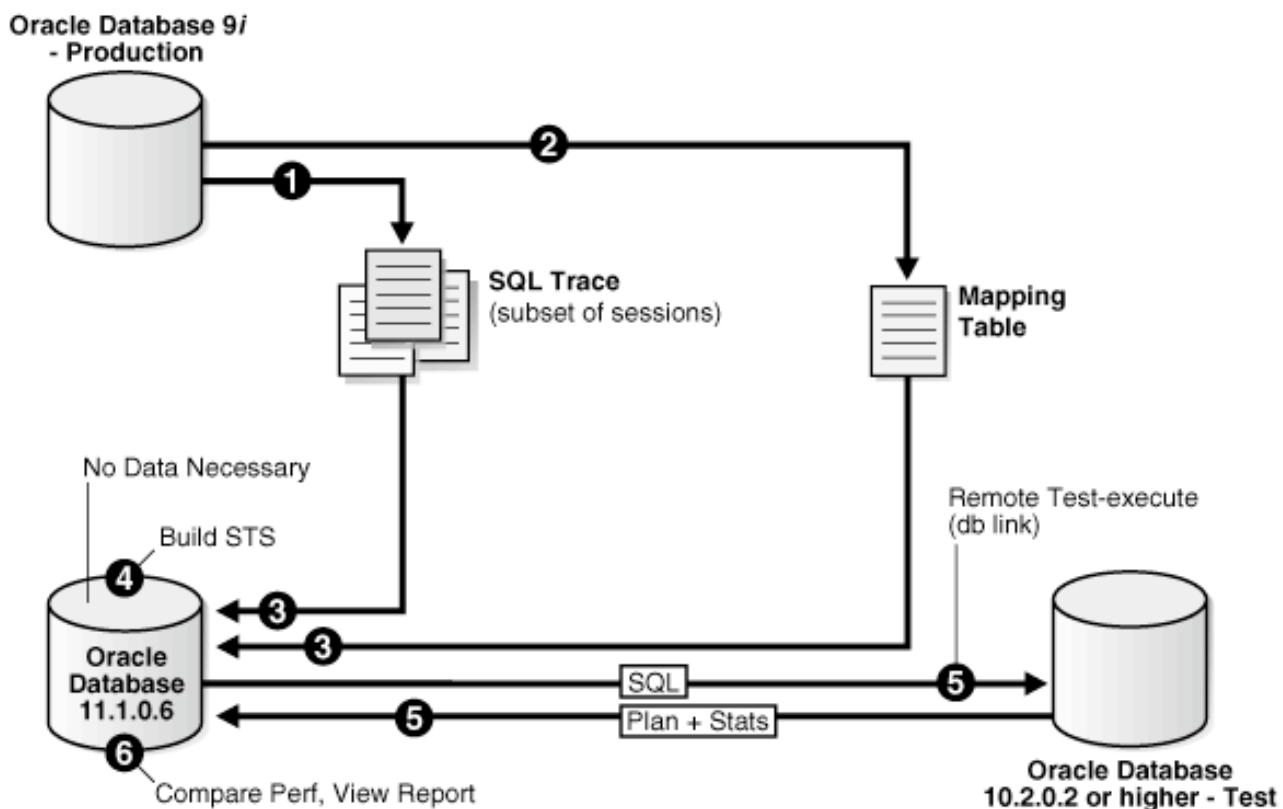


Seeding SPM – Cursor Cache

Please Resist the Temptation to go in Production with OFE set to a Previous Release!

You are going to have to change it at some point, better to figure it out when testing.

Seeding SPM – SQL Trace



Testing – More More Advanced Options

- **Import Original Stats**
 - **Then Generate New Stats**
 - **Use `dbms_stats.restore_schema_stats` to toggle**
 - **Use `dbms_stats.diff_table_stats_XXX` to compare**

*** Don't go into production with imported stats!
(you'll have to generate new stats sometime)**

And Finally!



References

Oracle White Paper – Oct 2009

SQL Plan Management in Oracle Database 11g

http://www.oracle.com/technology/products/bi/db/11g/pdf/twp_sql_plan_management_11gr2.pdf

Oracle White Paper – Oct 2009

Upgrading from Oracle Database 10g to 11g: What to expect from the Optimizer

http://www.oracle.com/technology/products/bi/db/11g/pdf/twp_upgrading_10g_to_11g_what_to_expect_from_optimizer.pdf

Oracle White Paper – Jul 2008

Upgrading from Oracle Database 9i to 10g: What to expect from the Optimizer

http://www.oracle.com/technology/products/bi/db/10g/pdf/twp_bidw_optimizer_10gr2_0208.pdf

Wolfgang Breitling – Mar 2008

Active Statistics (detailed testing of 11g auto_sample_size speed and accuracy)

<http://www.centrexcc.com/Active%20Statistics.ppt.pdf>



Questions / Contact Information

Questions?

Contact Info :

Kerry Osborne

kerry.osborne@enkitec.com

blog: kerryosborne.oracle-guy.com