The Best Oracle Database 11g New Features

Long Island, NY 2010

Rich Niemiec (rich@tusc.com), Rolta TUSC (www.rolta.com www.tusc.com)

(Thanks: Sridhar Avantsa, Mark Komine, Andy Mendelsohn, Debbie Migliore, Maria Colgan, Kamal Talukder. Steven Tyler, Roger Daltrey, Joe Perry, Aerosmith)

Audience Knowledge / Versions

- Oracle7 Experience ?
- Oracle8*i* Experience ?
- Oracle9*i* Experience ?
- Oracle10*g* Experience?
- Oracle Database 11*g* Experience?
- Oracle Database 11*g* R2 Experience?

- Goals
  - Present NEW features in an EASY way
  - Focus on a few nice features of Oracle11g

- Non-Goals
  - Learn ALL aspects of Oracle11g
  - Learn Tips that will make you money in Las Vegas

Overview

- Know the Oracle!
- Start Me Up – Using Memory Target, The Buffer Cache & The Result Cache
- Virtual Columns
- Invisible Indexes & Online Index Rebuilds
- DDL Lock Timeout, PL/SQL Expressions/Simple Integer
- Secure Files
- ADDM Enhancements
- SQL Plan Management (SPM) and capturing SQL Plan Baselines
- SQL Performance Analyzer, Access Advisor & Query Repair Advisor
- Exadata Simulation
- Real Application Testing (Database Capture and Replay) & Adding Nodes
- Interval Partitioning & Partition Compression
- Automatic Diagnostic Repository (ADR)
- Auto Sample, Creating Pending Statistics
- Adaptive Cursor Sharing and Bind Peeking
- EM, Grid Control, Security Enhancements & the Future Sizes

Know the Oracle

"I admire risk takers. I like leaders – people who do things before they become fashionable or popular. I find that kind of integrity inspirational."

LAWRENCE J. ELLISON | *Chairman & Chief Executive Officer, 2003*

Oracle Firsts – *Innovation!*

1979 First commercial SQL relational database management system
1983 First 32-bit mode RDBMS
1984 First database with read consistency
1987 First client-server database
1994 First commercial and multilevel secure database evaluations
1995 First 64-bit mode RDBMS
1996 First to break the 30,000 TPC-C barrier
1997 First Web database
1998 First Database - Native **Java** Support; Breaks 100,000 TPC-C
1998 First Commercial RDBMS ported to **Linux**
2000 First database with **XML**
2001 First middle-tier database cache
2001 First RDBMS with **Real Application Clusters**
2004 First **True Grid Database**
2005 First **FREE Oracle Database** (10g Express Edition)
2006 First **Oracle Support for LINUX Offering**
2007 **Oracle 11g Released!**
2008 Oracle Exadata Server Announced (Oracle buys BEA)
2009 Oracle buys Sun – Java; MySQL; Solaris; Hardware; OpenOffice
2010 Oracle announces MySQL Cluster 7.1, Exadata, Exalogic

2007: Version 11g was Released

- The Focus has been Acquisitions and gaining **Market Share**
- Oracle 11g Database extends an already large lead
  - Easier to Manage the Database – Better Grid Control
  - Self Tuning through a variety of tools (Makes 1 person equal 10)
  - Better Security/Encryption & Recoverability via Flashback
  - Better Testing Tools (Real Application Testing)
- Andy Mendelsohn is still the database lead
- Releases of **Siebel, PeopleSoft, JDE** and Oracle12 Apps.
- New Oracle BI Suite & **Acquisition of Hyperion**
- **Acquisition of BEA, SUN**

Oracle gets Sun: Java, MySQL, Solaris, OpenOffice, Hardware, Storage Tech

msnbc

featuring Today Show · Nightly News · Dateline · Meet the Press · MSN

ness / U.S. business

Oracle to buy Sun Microsystems for $7.4 billion

Deal comes after IBM abandoned its bid for networking equipment maker

AP Associated Press
updated 59 minutes ago

SAN FRANCISCO - Oracle Corp. snapped up computer server and software maker Sun Microsystems Inc. for $7.4 billion Monday, pouncing on an opportunity that opened up after rival IBM Corp. abandoned an earlier bid to buy one of Silicon Valley's best known — and most troubled — companies.

The deal will end Sun's 27-year history as Silicon Valley's brash independent and give Oracle ownership of the Java programming language, which runs on more than 1 billion devices around the world. Oracle also will take

Video

Launch

CNBC

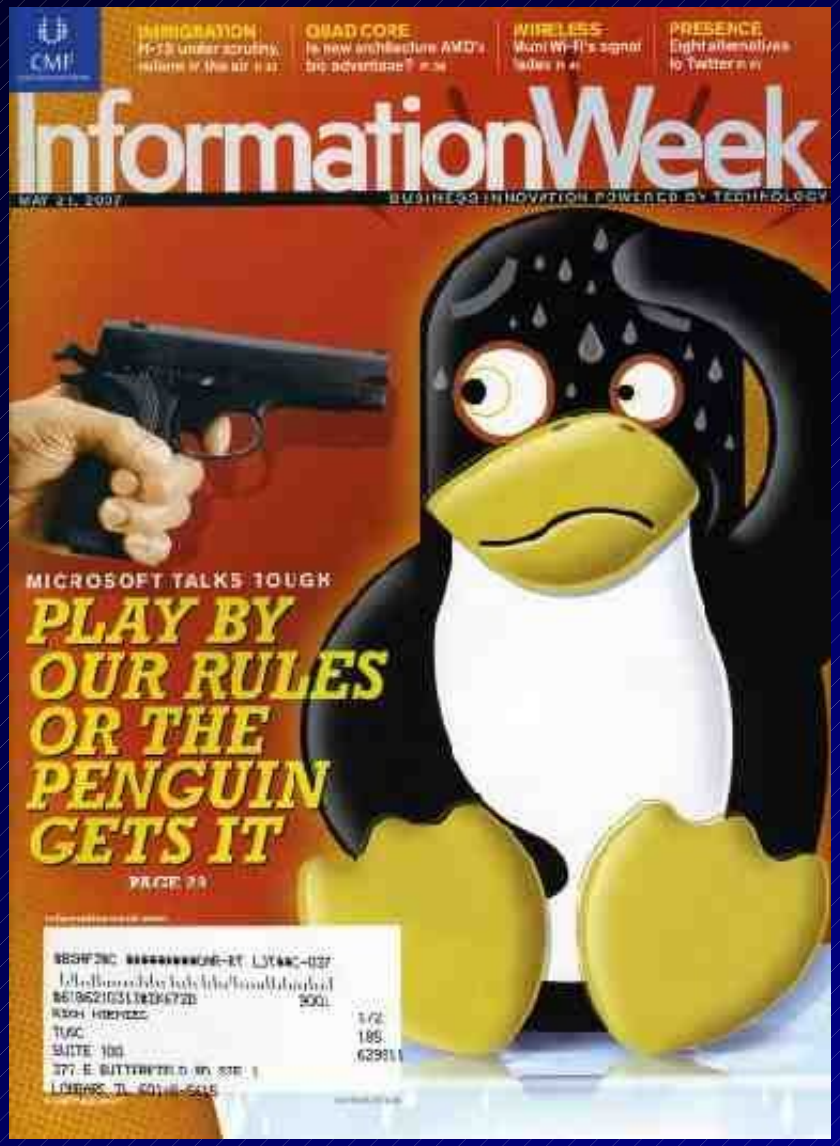
Most popular

Most viewed | Top rated

New Giant | A look at the two companies

	Oracle	Sun
Market Value*	$93.79 billion	$6.81 billion
Employees	86,500	30,000
FY08 Revenue	$22.43 billion	$13.88 billion
	Services $4.6; Database $11.6; Business software $6.2	Services $5.3; Servers $6.3; Storage $2.4
FYC0 Profit	$5.52 billion	$403 million
Key Products	Databases, business software from Siebel, PeopleSoft	Server computers, storage devices, Java and Solaris technology

*As of 4/20/09
Note: Sun's fiscal year ended 6/30/08; Oracle's fiscal year ended 5/31/08
Source: WSJ Market Data Group


CMP
IMMIGRATION
H-1B under scrutiny, reform in the air
QUAD CORE
Is new architecture AMD's big advantage?
WIRELESS
Muni Wi-Fi's signal fades
PRESENCE
Eight alternatives to Twitter
InformationWeek
MAY 21, 2007
BUSINESS INNOVATION POWERED BY TECHNOLOGY
MICROSOFT TALKS TOUGH
PLAY BY OUR RULES OR THE PENGUIN GETS IT
PAGE 24
TUSC
SUITE 100

Why will Oracle Win in the Future?
My Reasons... (Teach your kids Oracle!)

- It's not about the database anymore, now it's about the Applications. When it is about the database, security and high availability are issues where Oracle excels.
- Made acquisitions at the right time in the market
- Oracle has Great Developers who love what they do
- Oracle has Great Sales & Marketing that's maturing
- I believe Oracle will do well with or without Larry down the road, but Larry's creative mind and risk taking is not as easy to replicate in a Fortune 500 CEO. Other CEO's might be too willing to benefit Wall Street first versus benefit Oracle (long term). It would be a loss.

Testing the Future Version

Version 11.1.0.6.0 of the Database

Version 11.2.0.1.0 of the Database for Release 2 Examples

Oracle Database 11g Release 1: Upgrade Paths

Direct Upgrade Path

Source Database	Target Database
9.2.0.4.0 (or higher)	11.1.x
10.1.0.2.0 (or higher)	11.1.x
10.2.0.1.0 (or higher)	11.1.x

In-Direct Upgrade Path

Source Database	Upgrade Path for Target Database	Target Database
7.3.3.0.0 (or lower)	7.3.4.x --> 9.2.0.8	11.1.x
8.0.5.0.0 (or lower)	8.0.6.x --> 9.2.0.8	11.1.x
8.1.7.0.0 (or lower)	8.1.7.4 --> 9.2.0.8	11.1.x
9.0.1.3.0 (or lower)	9.0.1.4 --> 9.2.0.8	11.1.x

Database Upgrade Assistant (DBUA)

- Command Line Option to Auto Extend System Files
- Express Edition Upgrade to others
- Integration with Oracle Database 11g Pre-upgrade Tool
- Moving Data Files into ASM, SAN, and Other File Systems
- Oracle Base and Diagnostic Destination Configuration

Database Upgrade Assistant (DBUA)

- DBUA checks before the upgrade:
  - Invalid user accounts or roles
  - Invalid data types or invalid objects
  - De-supported character sets
  - **Adequate resources** (rollback segments, tablespaces, and free disk space)
  - Missing SQL scripts needed for the upgrade
  - Listener running (if Oracle Enterprise Manager Database Control upgrade or configuration is requested)
  - Oracle Database software linked with Database Vault option. If Database Vault is enabled, Disable Database Vault before upgrade.

The New Version – Life is Good!

$ sqlplus ***/***

SQL*Plus: Release 11.1.0.6.0 - Production on Tue Oct 30 11:21:04 2007
Copyright (c) 1982, 2007, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and **Real Application Testing options**

SQL> startup
ORACLE instance started.
Total System Global Area 422670336 bytes
Fixed Size 1300352 bytes
Variable Size 306186368 bytes
Database Buffers 109051904 bytes
Redo Buffers 6131712 bytes

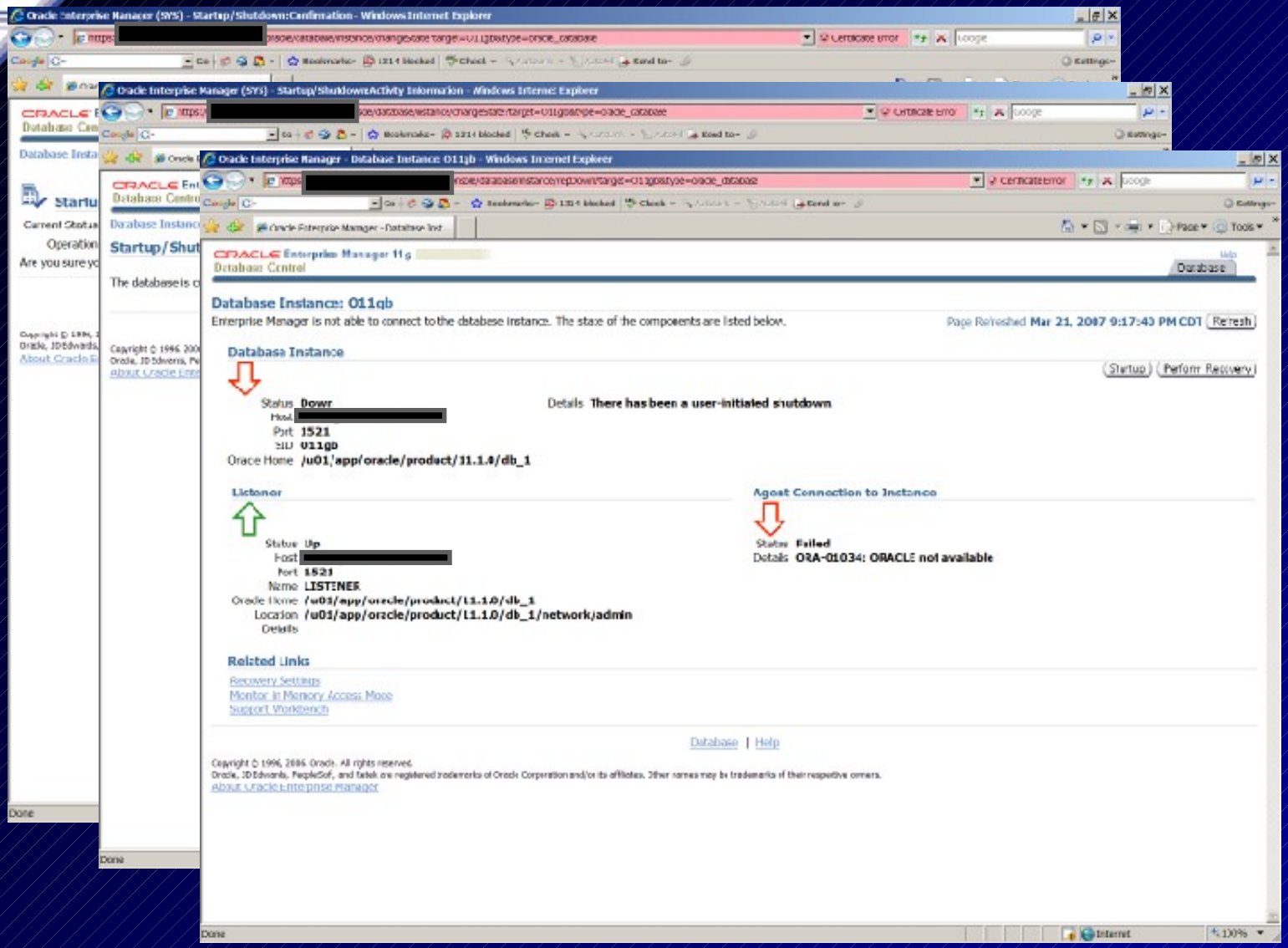
Database mounted.
Database opened.

```
SYS@sillgr2>
SYS@sillgr2> select * from v$version;

BANNER
--------------------------------------------------------------------------------
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
PL/SQL Release 11.2.0.1.0 - Production
CORE    11.2.0.1.0      Production
TNS for Linux: Version 11.2.0.1.0 - Production
NLSRTL Version 11.2.0.1.0 - Production

SYS@sillgr2>
SYS@sillgr2>
SYS@sillgr2>
SYS@sillgr2>
```

Or... Use 11g EM...


Startup/Shutdown:Specify Host and Target Database Credentials
Host Credentials
Database Credentials
Database

Or... Use 11g EM...Status...

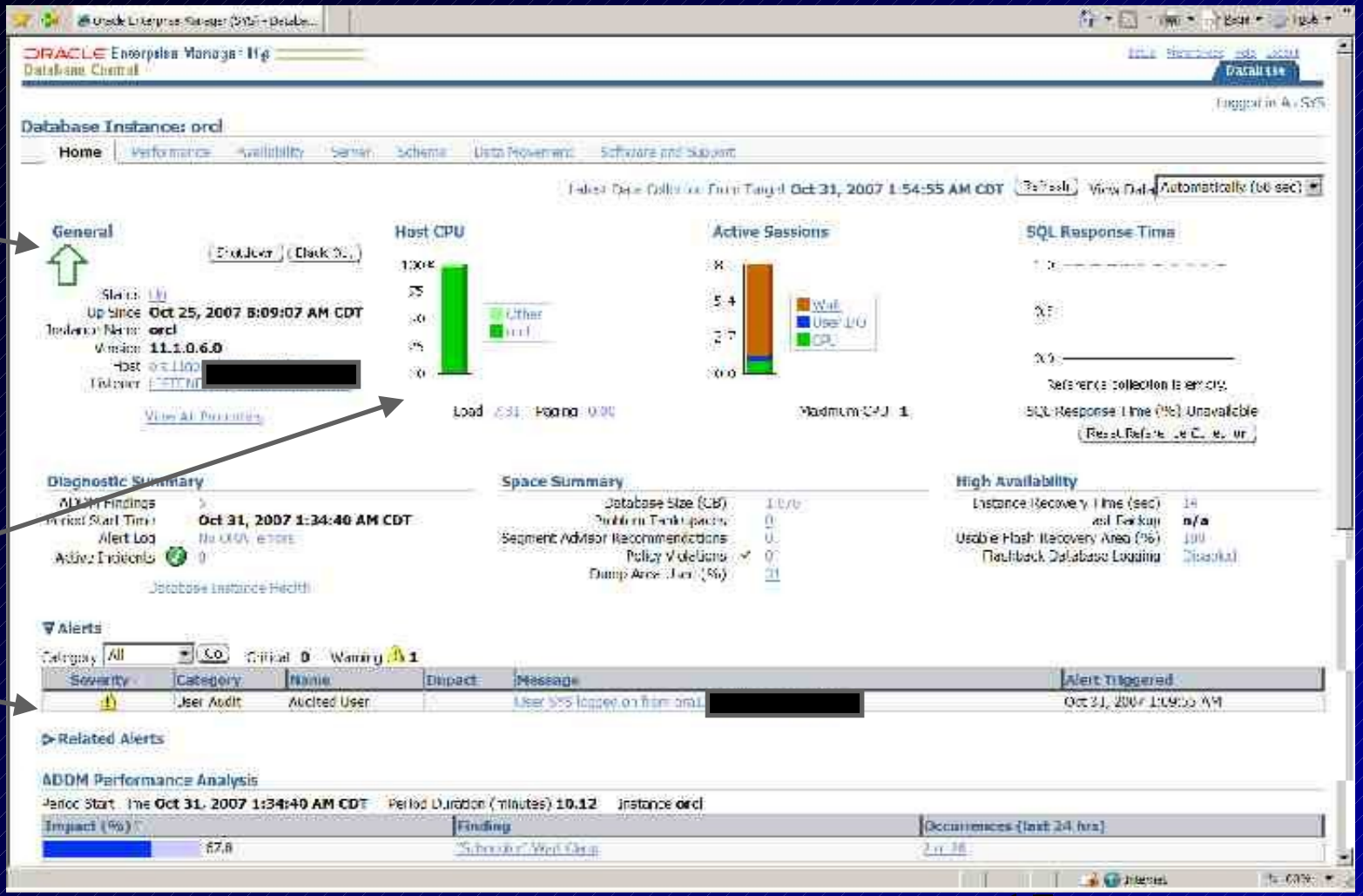


Database Information - UP!
11gR1

Monitor Database (UP)

Users are Definitely Using it!

We have an alert – we logged on as SYS



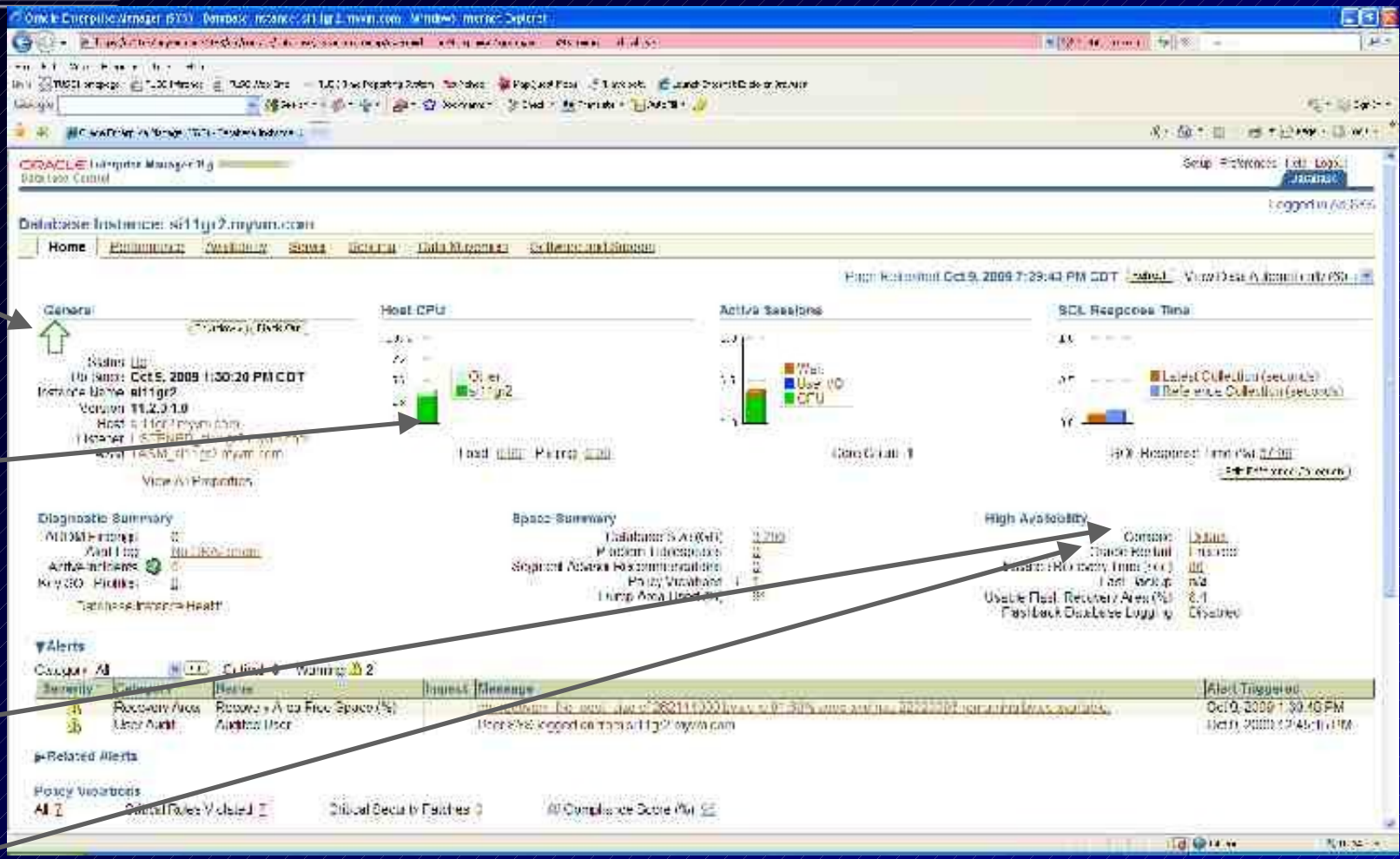
Database Information - UP!
11gR2

Monitor Database (UP)

Users are Using it!

Click on the HA Console

Restart Enabled

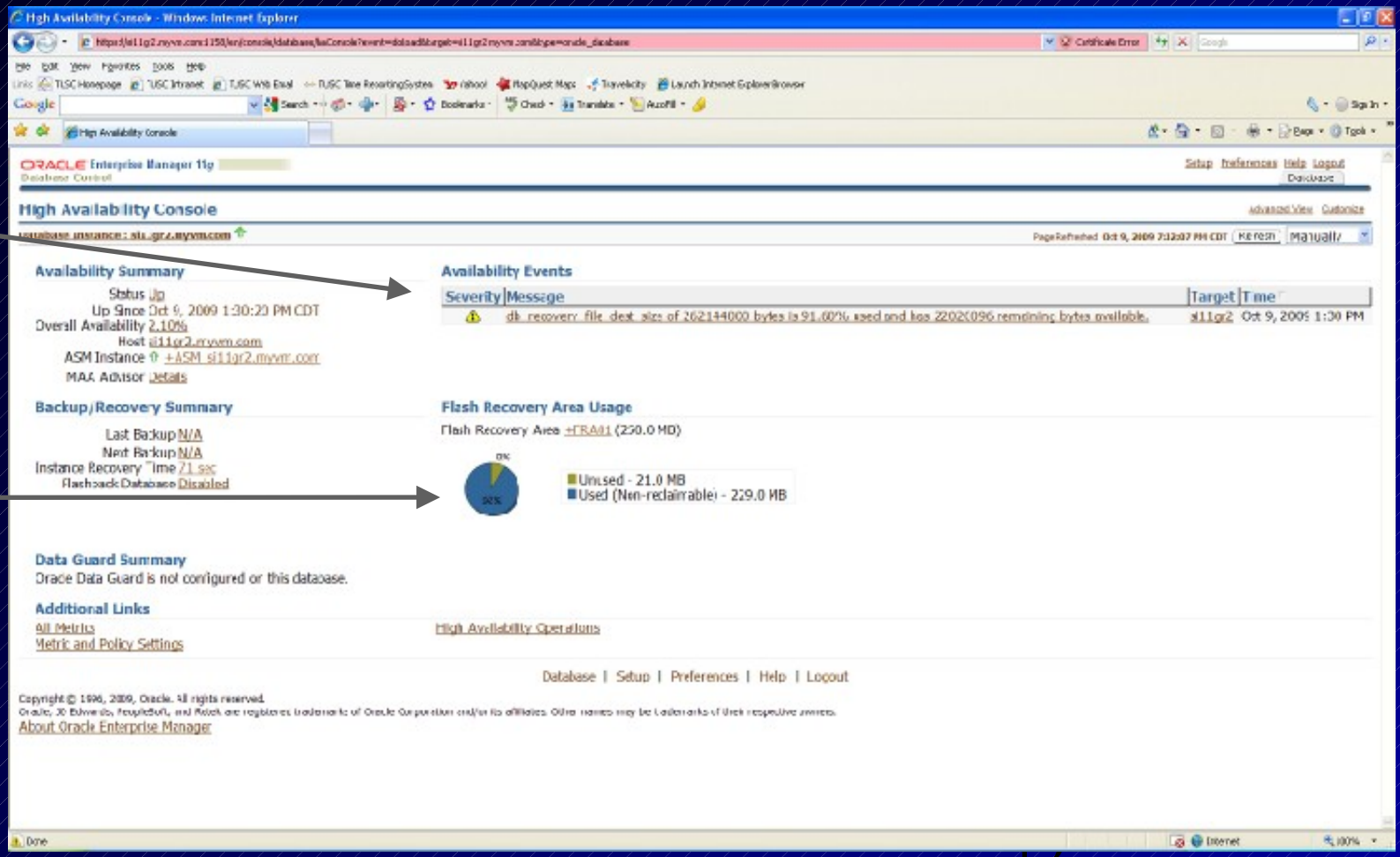


(Restart Database, ASM, Listener after restart of Software/Hardware)

Database Information - UP!
High Availability Console - 11gR2

Events that are an issue

Flash Recovery Usage

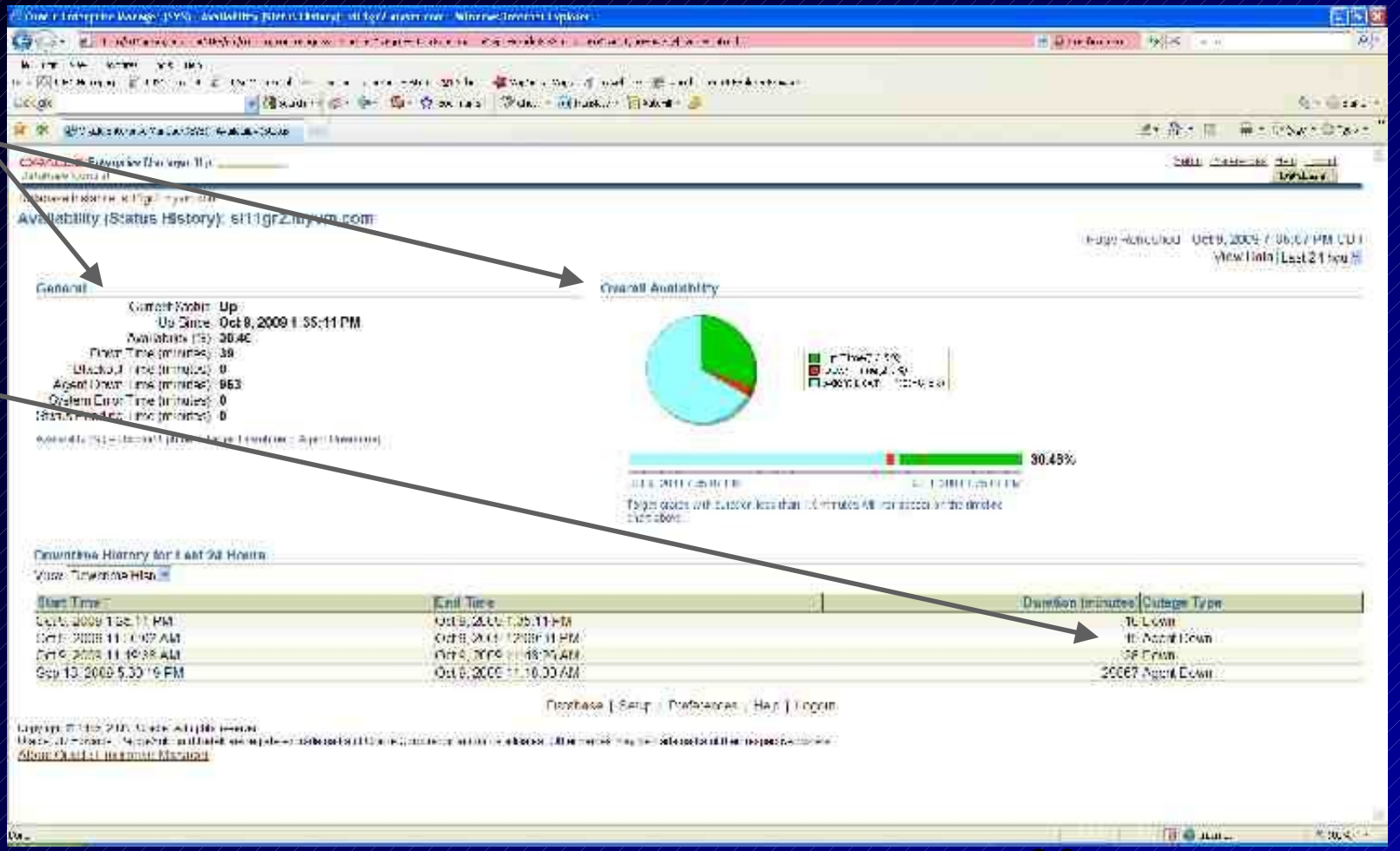


Database Information - UP!

HA - *Availability* - 11gR2

Up Time

Outage Type

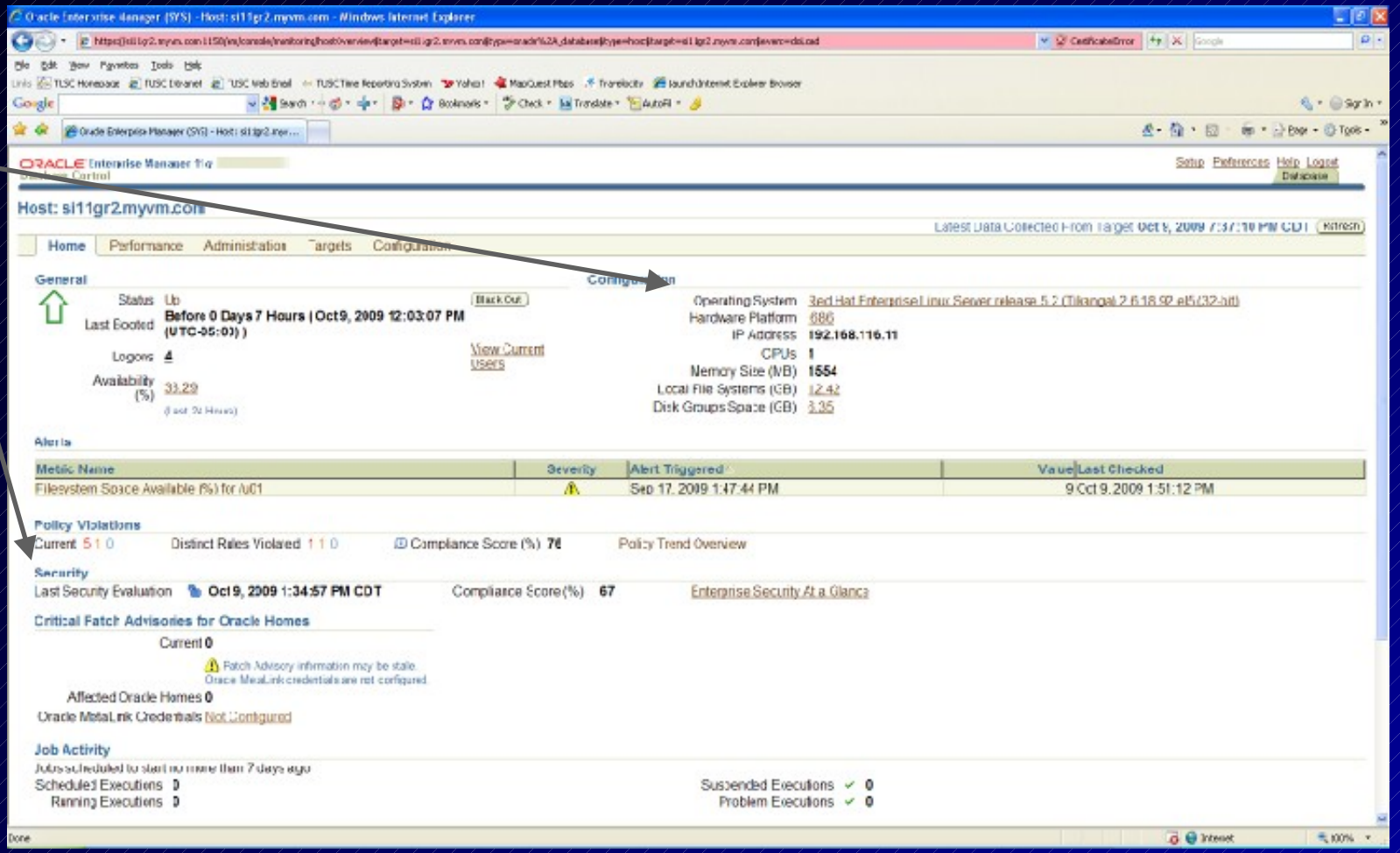


Database Information - UP!

HA *Host* - 11gR2

RH 5.2

Latest Security Evaluation

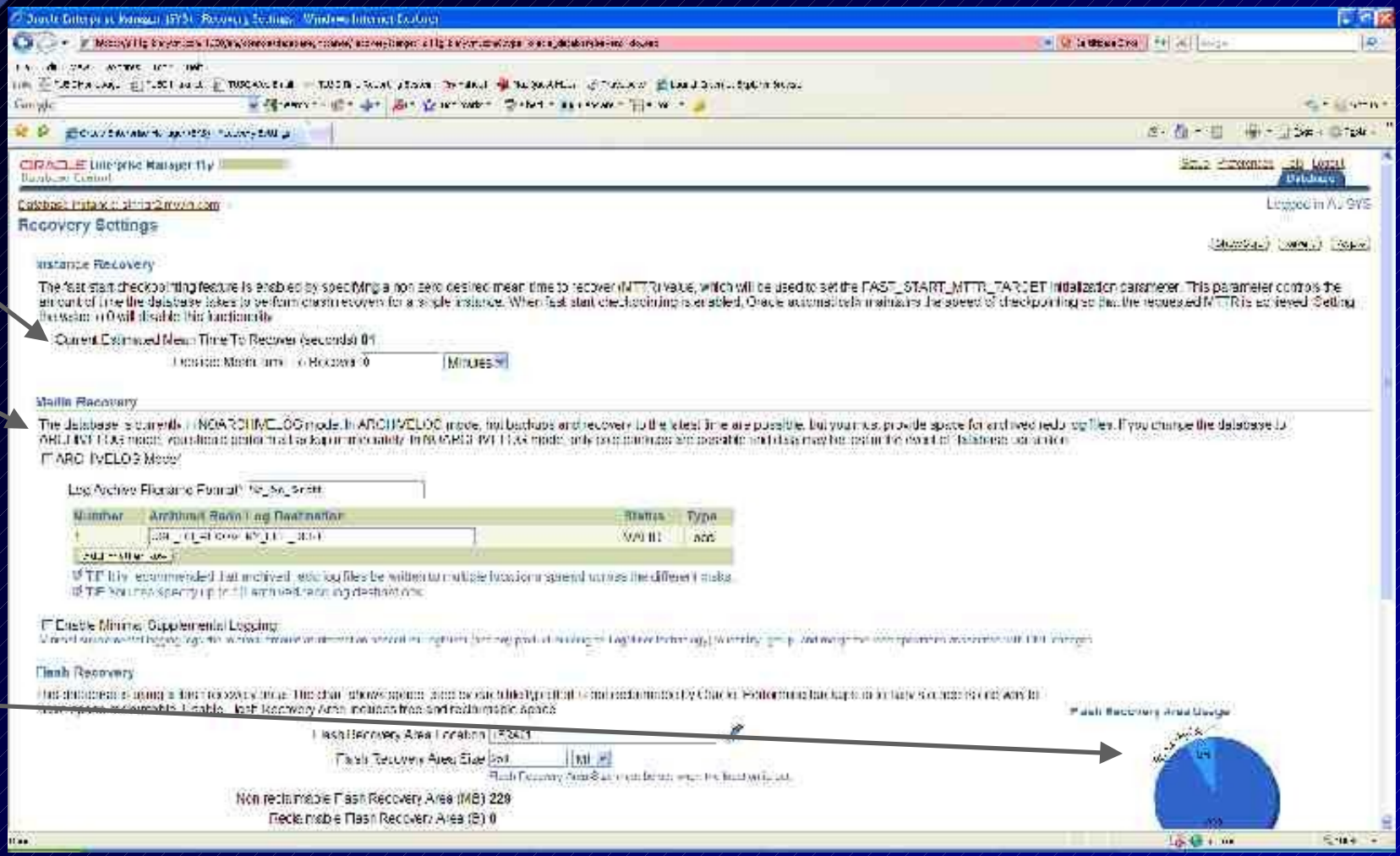


Database Information - UP!
HA *Recovery Settings* - 11gR2

Mean Time to Reco

Warns on ***noarch*** mode

Flash Reco Area Usage

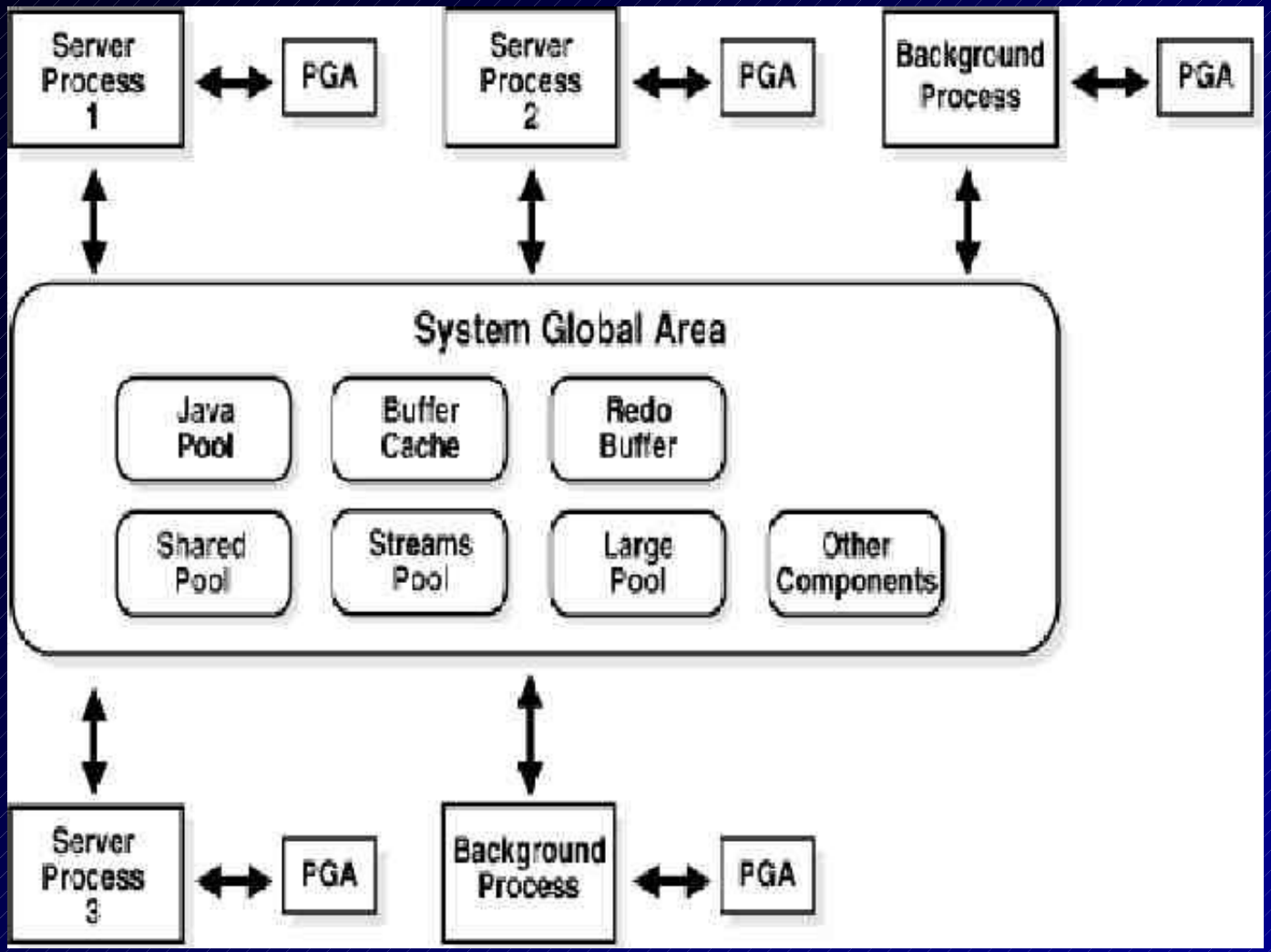


MEMORY_TARGET
&
Automatic Memory Management

Automatic Memory Management (AMM)
MEMORY_TARGET in 11g

- First there was some Automatic Memory Mgmt - 9i
  - SGA_MAX_SIZE introduced in 9i – Dynamic Memory
  - No more Buffers – DB_CACHE_SIZE
  - Granule sizes introduced - _ksm_granule_size
- Then came SGA_TARGET – 10g
  - Oracle Applications recommends setting this for SGA
  - Set minimums for key values (Data Cache / Shared Pool)
- Now there is MEMORY_TARGET – 11g
  - SGA + PGA all in one setting; Still set minimums

SGA & PGA will be MEMORY_TARGET



Automatically sized SGA Components that Use SGA_TARGET

Component	Initialization Parameter
Fixed SGA	None
Shared Pool	SHARED_POOL
Large Pool	LARGE_POOL_SIZE
Java Pool	JAVA_POOL_SIZE
Buffer Cache	DB_CACHE_SIZE
Streams Pool	STREAMS_POOL_SIZE

Manually Sized SGA Components that Use SGA_TARGET

Component	Initialization Parameter
Log buffer	LOG_BUFFER (pfile only in 10g)
Keep Pool	DB_KEEP_CACHE_SIZE
Recycle Pool	DB_RECYCLE_CACHE_SIZE
Block caches	DB_nK_CACHE_SIZE

Program Global Area (now in MEMORY_TARGET):

Aggregate PGA	PGA_AGGREGATE_TARGET

Automatic Memory Management (AMM)
MEMORY_TARGET in 11g


```
SQL> sho parameter sga_

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
sga_max_size                         big integer 360M
sga_target                           big integer 0

SQL> sho parameter memory

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
memory_max_target                    big integer 360M
memory_target                        big integer 360M
```

Moving from SGA_TARGET to: MEMORY_TARGET


```
SQL> sho parameter target

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
memory_max_target                    big integer 0
memory_target                        big integer 0
pga_aggregate_target                 big integer 110M
sga_target                           big integer 250M
```

Moving from SGA_TARGET to: MEMORY_TARGET

ALTER SYSTEM SET MEMORY_MAX_TARGET=360M SCOPE=SPFILE;
(shutdown/startup)
ALTER SYSTEM SET MEMORY_TARGET=360M SCOPE=SPFILE;
ALTER SYSTEM SET SGA_TARGET=0; (or set a minimum)
ALTER SYSTEM SET PGA_AGGREGATE_TARGET=0; (or set a minimum)

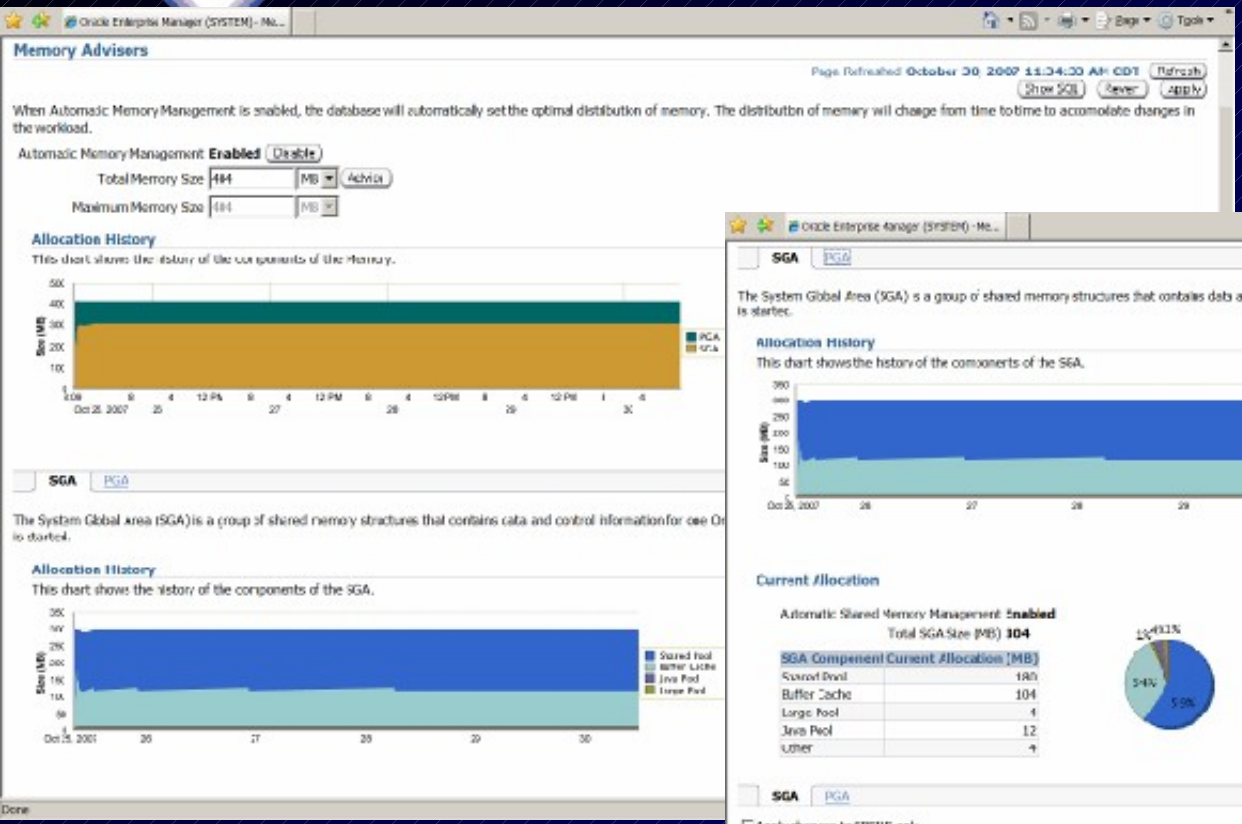
SQL> sho parameter target

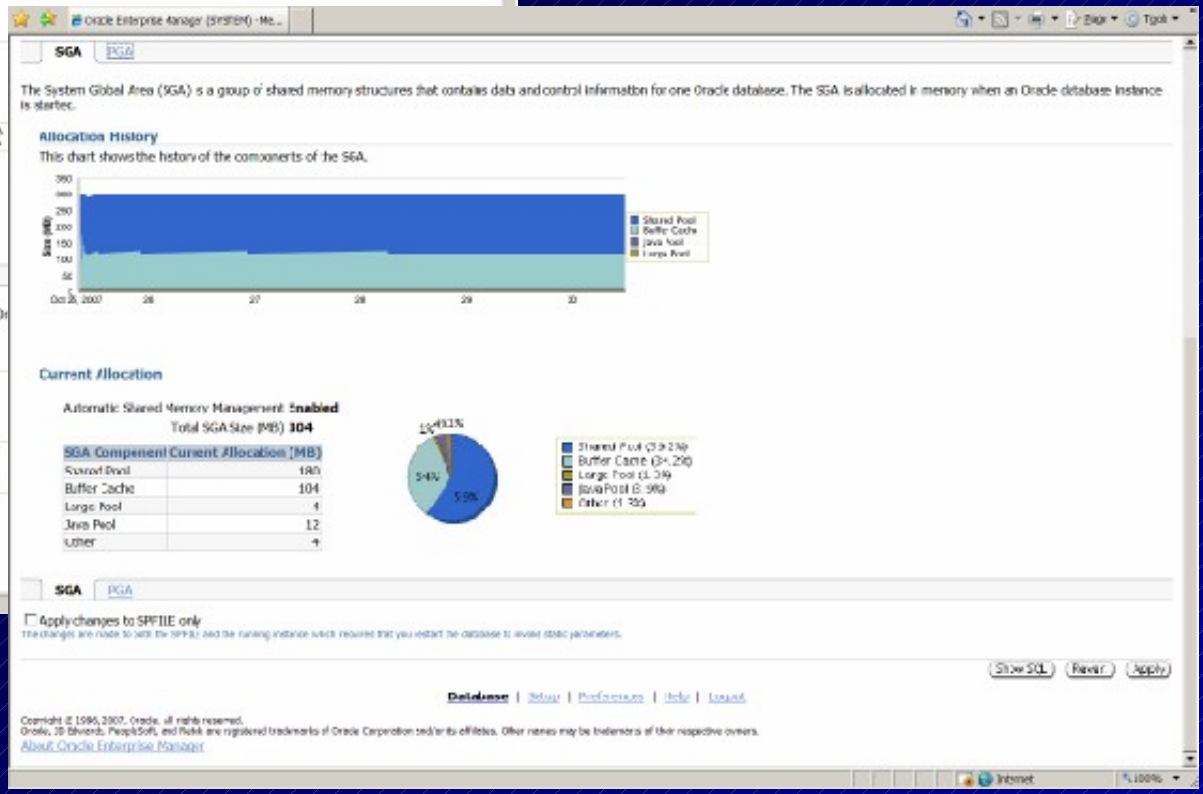
NAME	TYPE	VALUE
memory_max_target	big integer	360M
memory_target	big integer	360M
pga_aggregate_target	big integer	0
sga_target	big integer	0

Moving from SGA_TARGET to: MEMORY_TARGET (set minimums)

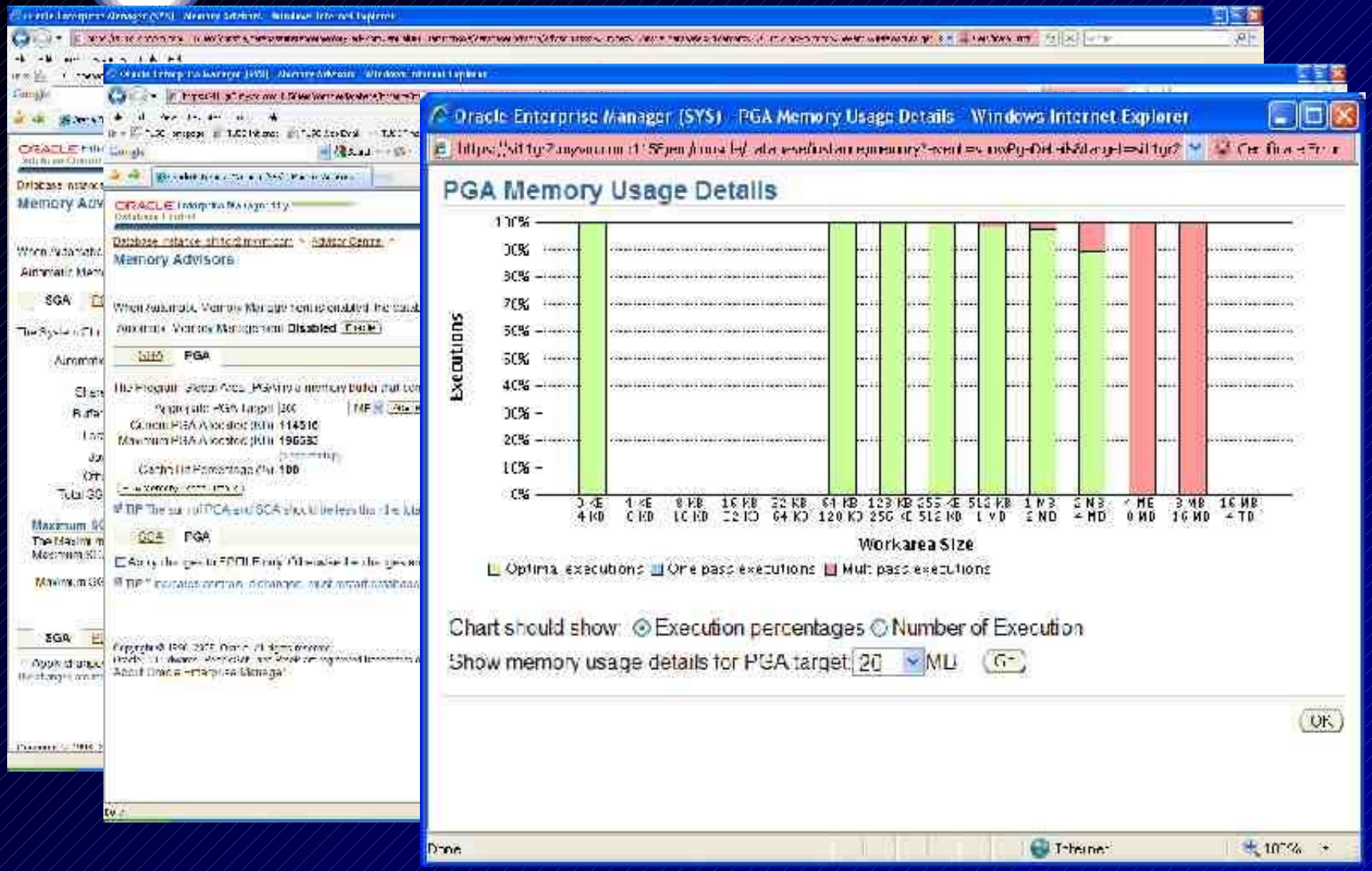

```
ALTER SYSTEM SET SGA_TARGET=200;
ALTER SYSTEM SET PGA_AGGREGATE_TARGET=100;

SQL> sho parameter target

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
memory_max_target                    big integer 360M
memory_target                        big integer 360M
pga_aggregate_target                 big integer 100M
sga_target                           big integer 200M
```

Moving from SGA_TARGET to: MEMORY_TARGET - EM





Moving from SGA_TARGET to:
Memory Advisor – 11gR2



Buffer Cache & Result Cache

First, A quick review: Flush Buffer Cache

- The new 10g feature allows the flush of the buffer cache. It is NOT intended for production use, but rather for system testing purposes.
- This can help you in your tuning needs or as a band-aid if you have 'free buffer' waits (there are better ways to fix this like writing more often or increasing the DB_CACHE_SIZE)
- Note that any Oracle I/O not done in the SGA counts as a physical I/O. If your system has O/S caching or disk caching, the actual I/O that shows up as physical may indeed be a memory read outside of Oracle.
- To flush the buffer cache perform the following:

```
SQL> ALTER SYSTEM FLUSH BUFFER_CACHE;
```

Flush Buffer Cache Example


```
select count(*) from tab1;

  COUNT(*)
----------
      1147


Execution Plan
----------------------------------------------------------------------------------------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE (Cost=4 Card=1)
   1    0   SORT (AGGREGATE)
   2    1     TABLE ACCESS (FULL) OF 'TAB1' (TABLE) (Cost=4 Card=1147)


Statistics
----------------------------------------------------
          0  db block gets
          7  consistent gets
          6  physical reads
```

Flush Buffer Cache Example

select count(*) from tab1; **(Run it again and the physical reads go away)**

```
  COUNT(*)
----------
      1147

Execution Plan
----------------------------------------------------------------------------------------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE (Cost=4 Card=1)
   1    0   SORT (AGGREGATE)
   2    1     TABLE ACCESS (FULL) OF 'TAB1' (TABLE) (Cost=4 Card=1147)

Statistics
----------------------------------------------------
          0  db block gets
          7  consistent gets
          0  physical reads
```

Flush Buffer Cache Example


```
ALTER SYSTEM FLUSH BUFFER_CACHE;
System altered.
```

select count(*) from tab1; **(Flush the cache and the physical reads are back)**

```
  COUNT(*)
----------
      1147

Execution Plan
----------------------------------------------------------------------------------------------------------------------
   0      SELECT STATEMENT Optimizer=CHOOSE (Cost=4 Card=1)
   1    0   SORT (AGGREGATE)
   2    1     TABLE ACCESS (FULL) OF 'TAB1' (TABLE) (Cost=4 Card=1147)

Statistics
----------------------------------------------------------
          0  db block gets
          7  consistent gets
          6  physical reads
```

The Result Cache

- Function Results of queries and query fragments can be cached in memory for **future executions**.
  - Choose calculations that frequently run
  - Choose data that does NOT frequently change
- **RESULT_CACHE** & RELIES_ON clauses
- **Takes its memory from the Shared Pool**
  - Set with RESULT_CACHE_SIZE
  - **RESULT_CACHE_MODE=force** (auto/**manual**)
- DBMS_RESULT_CACHE.FLUSH to clear
- Is NOT passed between RAC/Grid nodes
- Check the docs for other Restrictions & Rules!!

Result Cache Performance Example Query (1M Row Test)


```
select *
from (select *
        from (select t.country_name, t.city_name,
                sum(t.salary) a_sum, max(t.salary) a_max
              from emps t
              group by t.country_name, t.city_name)
              order by a_max desc)
where rownum < 2;
```

Result Cache
Example Performance

Step 1 - In Session 1-

Executed query without hint and it returned an elapsed time of 3.80 seconds (not cached).

Step 2 - In Session 2 –

Executed query without hint and it returned an elapsed time of 3.20 seconds (not cached).

Result Cache Example Performance

Step 3 - In Session 2

Executed query with the RESULT_CACHE hint and it returned an elapsed time of 3.18 seconds (cache it).

Step 4 - In Session 1

Executed query without the RESULT_CACHE hint, but with **RESULT_CACHE_MODE=force** and it returned an elapsed time of 0.86 seconds (cached!!).

Result Cache Example Query From the Oracle Docs

- The RELIES_ON Clause specifies tables or views that the Function Results are dependent on.

```
-- Package specification
CREATE OR REPLACE PACKAGE HR IS
   ...
   type DeptInfoRec IS RECORD (avgSal NUMBER,
                               numberEmployees NUMBER);
-- Function declaration
   FUNCTION GetDeptInfo (dept_id NUMBER) RETURN DeptInfoRec
   RESULT_CACHE;
   ...
END HR;
```

Result Cache Example Query From the Oracle Docs

```
PACKAGE BODY HR IS
  ...
  -- Function definition
  FUNCTION GetDeptInfo (dept_id NUMBER) RETURN DeptInfoRec
      RESULT_CACHE RELIES_ON (EMP);
  IS
      result DeptInfoRec;
  BEGIN SELECT AVG(sal), count(*) INTO result
                FROM EMP
              WHERE deptno = dept_id;
             RETURN result;
  END;
  ...
END HR;
```

The Result Cache – V$ Views

- V$RESULT_CACHE_STATISTICS – Displays the amount of memory to help you determine memory currently allocated to the result cache.

Other V$ views:

- V$RESULT_CACHE_MEMORY
- V$RESULT_CACHE_OBJECTS
- V$RESULT_CACHE_DEPENDENCY

The Result Cache – FYI Only Digging Deeper

KSPPINM	KSPPSTVL	KSPPDESC
_result_cache_auto_execution_threshold	**1**	**result cache auto execution threshold**
_result_cache_auto_size_threshold	100	result cache auto max size allowed
_result_cache_auto_time_threshold	1000	result cache auto time threshold
_result_cache_block_size	**1024**	**result cache block size**
_result_cache_bypass	**FALSE**	**bypass the result cache**
_result_cache_hash_buckets	**1024**	**hash bucket count**
_result_cache_invalid	0	post-invalidation usage allowance
_result_cache_max_result	100	maximum result size as percent of cache size
_result_cache_remote_expiration	0	maximum life time (min) for any result using a remote object
_result_cache_timeout	60	maximum time (sec) a session waits for a result

Tuning Tools – FYI Only
DBMS_XPLAN

- Use DBMS_XPLAN to query the execution plan
  - Automatically queries the last plan in PLAN_TABLE
  - uses a TABLE() function with another pipelined function
  - Operation text truncation might be a problem
  - Will give additional information after plan
    - Highlight filter vs join conditions, if plan table is current
    - Displays warning message of old version plan table is being used
  - In 11g, a procedure for SQL Plan Baselines (we'll cover these later).

```
DBMS_XPLAN.DISPLAY_SQL_PLAN_BASELINE (
      sql_handle   IN VARCHAR2 := NULL,
      plan_name    IN VARCHAR2 := NULL,
      format       IN VARCHAR2 := 'TYPICAL')   <'BASIC'|'ALL'>
RETURN dbms_xplan_type_table;
```

Tuning Tools – FYI Only
DBMS_XPLAN

DBMS_XPLAN Example:

Select *

from table (dbms_xplan.display);

```
PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------


-------------------------------------------------------------------------------------------
| Id  | Operation             |  Name         | Rows  | Bytes | Cost  | Pstart| Pstop |
-------------------------------------------------------------------------------------------
|   0 | UPDATE STATEMENT      |               |   328 |  2296 |     2 |       |       |
|   1 |  UPDATE               | JOURNAL_LINE  |       |       |       |       |       |
|   2 |   PARTITION RANGE ALL|               |       |       |       |     1 |     4 |
|   3 |    TABLE ACCESS FULL  | JOURNAL_LINE  |   328 |  2296 |     2 |     1 |     4 |
-------------------------------------------------------------------------------------------
Note: cpu costing is off, 'PLAN_TABLE' is old version
11 rows selected
```

The Virtual Column

The Virtual Column

- The value of the virtual column is a derived expression.
  - Can be derived from columns of the same table or from constants
  - Can include SQL or user-defined PL/SQL functions
- Virtual column DATA is NOT PHYSICALLY STORED.
- You CAN NOT explicitly write to a virtual column
- You CAN create a PHYSICAL index (result is function-based index) or partition on a virtual column <unlike a computed column in SQL Server or other databases>
- If you UPDATE columns of a virtual column and it has an index, then it will be computed on the UPDATE vs. on the SELECT (very important from a tuning standpoint).
- Index Organized and External Tables can NOT have virtual columns.

The Virtual Column

```
create table emp_rich
  (empno number(4),
   sal    number(7,2),
   yearly_sal generated always as (sal*12),
   deptno number(2));
```

Table created.

```
insert into emp_rich(empno, sal, deptno)
            select empno, sal, deptno from scott.emp;
```

14 rows created.

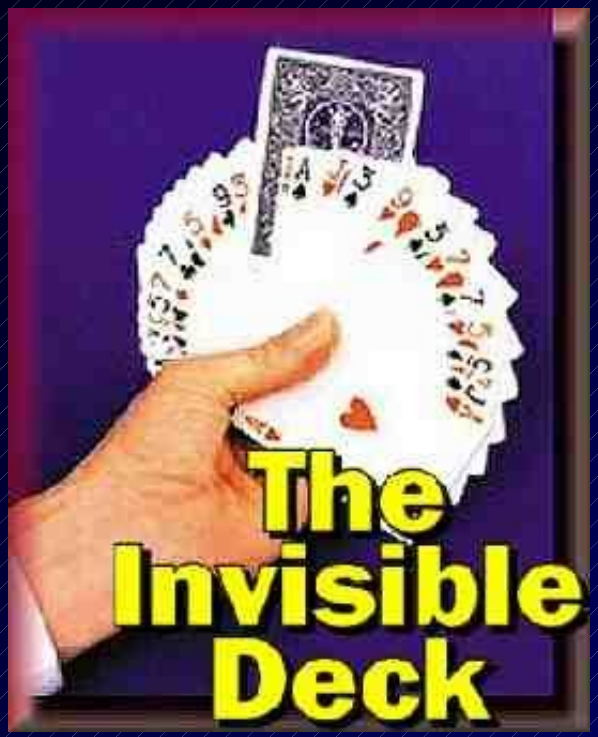
The Virtual Column

select * from emp_rich;

EMPNO	SAL	YEARLY_SAL	DEPTNO
7369	800	9600	20
7499	1600	19200	30
7521	1250	15000	30
7566	2975	35700	20
7654	1250	15000	30
7698	2850	34200	30

...

The Invisible Index



The Invisible Index

- Set an index to VISIBLE or INVISIBLE
  - ALTER INDEX idx INVISIBLE;
  - ALTER INDEX idx VISIBLE;
  - CREATE INDEX... INVISIBLE;
- Great to turn off indexes for a while when you think they're not being used, but BEFORE you drop them.
- Can use INDEX (to override invisibility) or NO_INDEX (to override visibility) hints to override either setting.
- **The index IS MAINTAINED during DML**
- Great for testing!

The Invisible Index

create index deptno_invisible_idx on dept_rich(deptno) invisible;

Index created.

select count(*) from dept_rich where deptno = 30; (doesn't see the index)

```
COUNT(*)
----------
       512

Execution Plan
----------------------------------------------------------
Plan hash value: 3024595593

-------------------------------------------------------------------------------
| Id  | Operation          | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |           |     1 |     2 |     4   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE    |           |     1 |     2 |            |          |
|*  2 |   TABLE ACCESS FULL| DEPT_RICH |   512 |  1024 |     4   (0)| 0:00:01 |
-------------------------------------------------------------------------------
```

The Invisible Index

select /*+ index(dept_rich dept_rich_inv_idx) */ count(*) from dept_rich where deptno = 30; (forces the index with hint)

```
COUNT(*)
-------------
        512

Execution Plan
----------------------------------------------------------
Plan hash value: 3699452051
-------------------------------------------------------------------------------------
| Id  | Operation          | Name               | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |                    |     1 |     2 |     1   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE    |                    |     1 |     2 |            |          |
|*  2 |   INDEX RANGE SCAN| DEPT_RICH_INV_IDX |   512 |  1024 |1 (0)| 00:00:01
    |
-------------------------------------------------------------------------------------
```

The Invisible Index (set visible)

alter index dept_rich_inv_idx visible;

Index altered.

select count(*) from dept_rich where deptno = 30;

(it does see the index)

```
COUNT(*)
----------
       512

Execution Plan
----------------------------------------------------------
Plan hash value: 3699452051
--------------------------------------------------------------------------------
| Id  | Operation         | Name              | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |                   |     1 |     2 |     1   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE   |                   |     1 |     2 |            |          |
|*  2 |   INDEX RANGE SCAN| DEPT_RICH_INV_IDX |   512 |  1024 |1 (0)| 00:00:01 |
--------------------------------------------------------------------------------
```

The Invisible Index (set visible)

select /*+ no_index(dept_rich dept_rich_inv_idx) */ count(*)
from dept_rich
where deptno = 30; (forces *not* using the index with hint)

```
  COUNT(*)
---------
       512

Execution Plan
----------------------------------------------------------
Plan hash value: 3024595593

-------------------------------------------------------------------------------
| Id  | Operation          | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |           |     1 |     2 |     4   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE    |           |     1 |     2 |            |          |
|*  2 |   TABLE ACCESS FULL| DEPT_RICH |   512 |  1024 |     4   (0)| 0:00:01 |
-------------------------------------------------------------------------------
```

The Invisible Index (check it)

alter index dept_rich_inv_idx invisible;

Index altered.

select index_name, visibility
from dba_indexes (or go to USER_INDEXES)
where index_name = 'DEPT_RICH_INV_IDX';

INDEX_NAME	VISIBILITY
DEPT_RICH_INV_IDX	INVISIBLE

Create & Rebuild Index Online

Create & Rebuild Index Online

- You can create/rebuild indexes even when doing DML on the base table, but it's better to do during low DML activity.
- **Prior to Oracle 11g**, this required an exclusive lock at the beginning and end of the rebuild. This lock could cause DML delays and performance spike. This lock is no longer required for this operation.
- Rebuild is faster than a DROP and CREATE
- Basic Syntax:

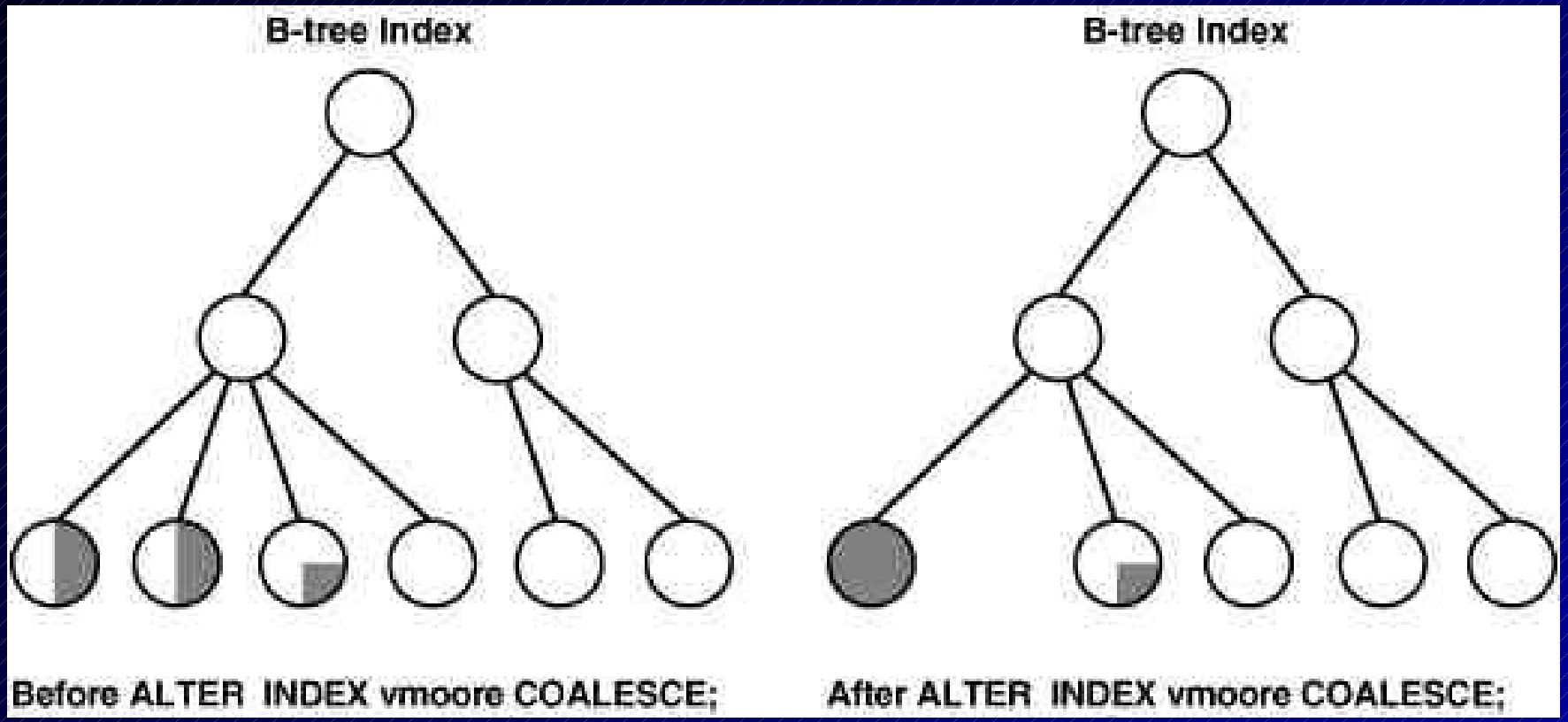
CREATE INDEX *index_name* ON *table (col1,...)* ONLINE;

Index created.

ALTER INDEX *index_name* REBUILD ONLINE;

Index altered.

Rebuild Index or Coalesce (FYI)
Coalesce Example from Oracle Doc.



Rebuild Index or Coalesce

Rebuild:

- Quickly move index to another tablespace
- Requires more disk space
- Creates new index tree and shrinks heights
- Change storage/tblspc w/o dropping

Coalesce

- Can't move to another tablespace
- Requires much less space than rebuild
- Coalesces leaf blocks that are in the same branch
- Quickly frees index leaf blocks for use

Nice Developer Tools/Improvements

DDL_LOCK_TIMEOUT

PL/SQL Expressions

Simple Integer

New PL/SQL Packages

The DDL Lock Timeout

- DDL Statements (Create/Alter/Drop) require exclusive locks and thus sometimes fail due to bad timing.
- The parameter DDL_LOCK_TIMEOUT specifies the amount of time (in seconds) the DDL statement will wait for the lock before timing out and failing.
- The default value is 0, the max value is 100000 (27.77 hours).
- Example:

```
alter session set DDL_LOCK_TIMEOUT = 30
```

Session altered.

Allow Sequences in PL/SQL Expressions

- In Previous Versions needed to retrieve the value of a sequence (CURRVAL / NEXTVAL) by invoking a cursor (explicit or implicit).

In 11g:

- **No cursor is needed** so the code is more efficient.
- For big jobs – Saves MANY cursors

Allow Sequences in PL/SQL Expressions

OLD Way

```
DECLARE
      V_NEW_VAL  NUMBER;
BEGIN
  SELECT MY_SEQ.NEXTVAL INTO V_NEW_VAL
    FROM DUAL;
END;
```

NEW Way

```
DECLARE
  V_NEW_VAL  NUMBER;
BEGIN
  V_NEW_VAL := MY_SEQ.NEXTVAL;
END;
```

Simple Integer Data Type

- Oracle added the new SIMPLE_INTEGER data type to be more efficient than PLS_INTEGER since the operations are done directly at the hardware level. There is also a built-in NOT NULL condition for SIMPLE_INTEGER.
- The performance is larger when the PLS_CODE_TYPE='NATIVE' vs. INTERPRETED
- We used a PL/SQL Block to loop through 1 million times incrementing a numeric variable by one. We executed the test for each of these three times.

Results:

NUMBER: 1.26s

PLS_INTEGER: 0.88s

SIMPLE_INTEGER: 0.65s

Additional Enhancements
New PL/SQL Packages

- DBMS_AUTO_TASK_ADMIN
- DBMS_COMPARISON
- DBMS_DG
- DBMS_EDITIONS_UTILITIES
- DBMS_HM (Health Monitor)
- DBMS_HPROF
- DBMS_MGD_ID_UTL
- DBMS_NETWORK_ACL_ADMIN
- DBMS_RESCONFIG
- DBMS_RESULT_CACHE
- DBMS_SQLDIAG (SQL Repair)
- DBMS_WORKLOAD_CAPTURE
- DBMS_WORKLOAD_REPLAY
- DBMS_XA
- DBMS_XDBADMIN
- DBMS_XEVENT
- DBMS_XMLDTD
- DBMS_XMLINDEX
- DBMS_XMLTRANSLATIONS
- SDO_RDF
- SDO_RDF_INFERENCE

Additional Enhancements
Enhanced PL/SQL Packages

- DBMS_ADVISOR
- DBMS_APPLY_ADM
- DBMS_AQ
- DBMS_AQADM
- DBMS_CAPTURE_ADM
- DBMS_CDC_PUBLISH
- DBMS_CDC_SUBSCRIBE
- DBMS_CQ_NOTIFICATION
- DBMS_DATA_MINING
- DBMS_DATA_MINING_TRANSFORM
- DBMS_DATAPUMP
- DBMS_EXPFIL
- DBMS_FLASHBACK
- DBMS_HS_PASSTHROUGH
- DBMS_LOB
- DBMS_LOGSTDBY
- DBMS_MGWADM
- DBMS_MVIEW
- DBMS_PREDICTIVE_ANALYTICS
- DBMS_RESOURCE_MANAGER
- DBMS_RLMGR
- DBMS_RULE_ADM
- DBMS_SCHEDULER
- DBMS_SERVER_ALERT
- DBMS_SESSION
- DBMS_SPACE
- DBMS_SQL
- DBMS_SQLTUNE
- DBMS_STATS
- DBMS_STREAMS_ADM
- DBMS_TRACE
- DBMS_UTILITY
- DBMS_WORKLOAD_REPOSITORY
- DBMS_XDB
- DBMS_XMLSCHEMA
- DBMS_XPLAN
- UTL_INADDR
- UTL_RECOMP
- UTL_SMTP
- UTL_TCP

Nice DBA Tool

Oracle Secure Files

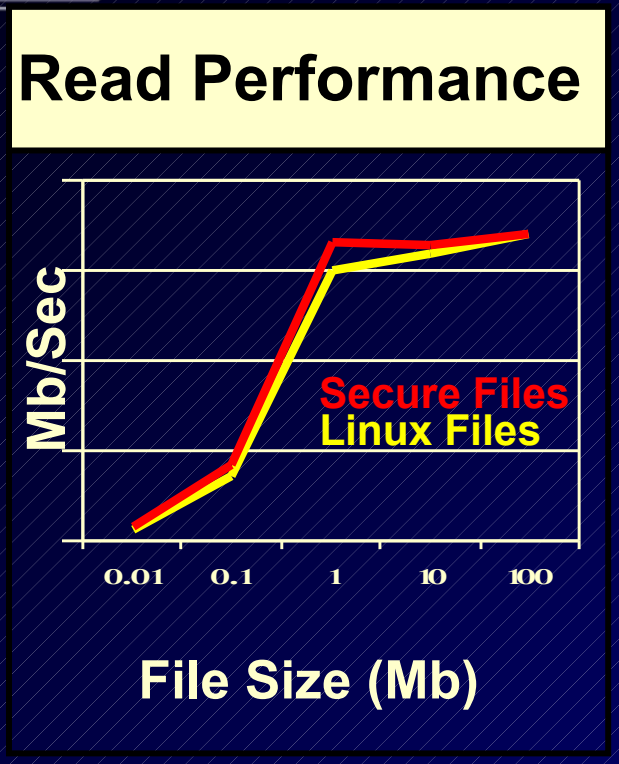
Oracle SecureFiles

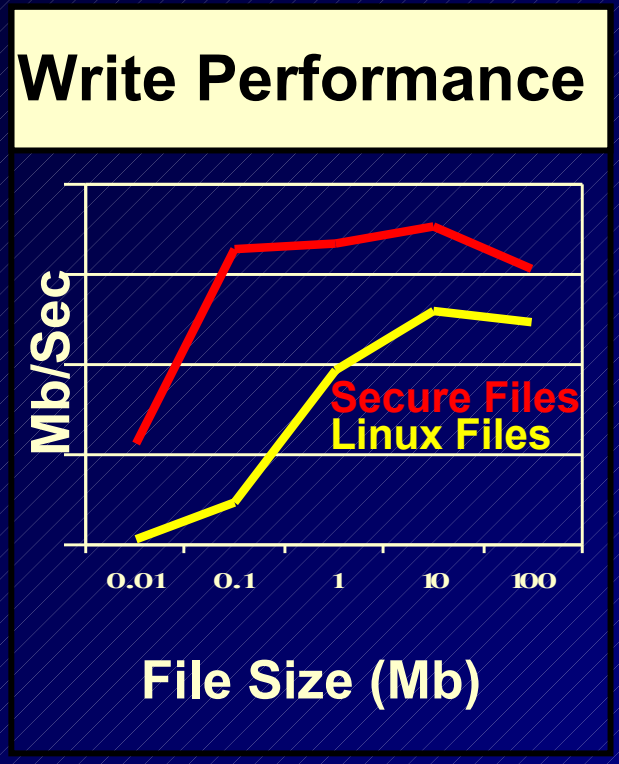
High-Performance Large Objects

- High-performance transactional access to large object data
  - RFID, DICOM medical, CAD, images, 3D spacial
  - low-latency, high throughput, concurrent access
  - space-optimized storage
- Protect your valuable data ... Keep large objects in the database!
  - transactions
  - transparent encryption
  - compression and de-duplication
  - database-quality security, reliability, and scalability
- Better security, single view and management of data
- Superset of LOB interfaces – easy migration

Oracle Secure Files

Better Performance than LOBs…





Adding Files using New Disk Space – 2x fast than LOBs

Adding Files using Deleted Space – 22x faster than LOBs

PL/SQL Reads – 6x Faster than LOBs

Your mileage will vary....

ADDM Enhancements
(Automatic Database Diagnostic Monitor)

ADDM enhancements

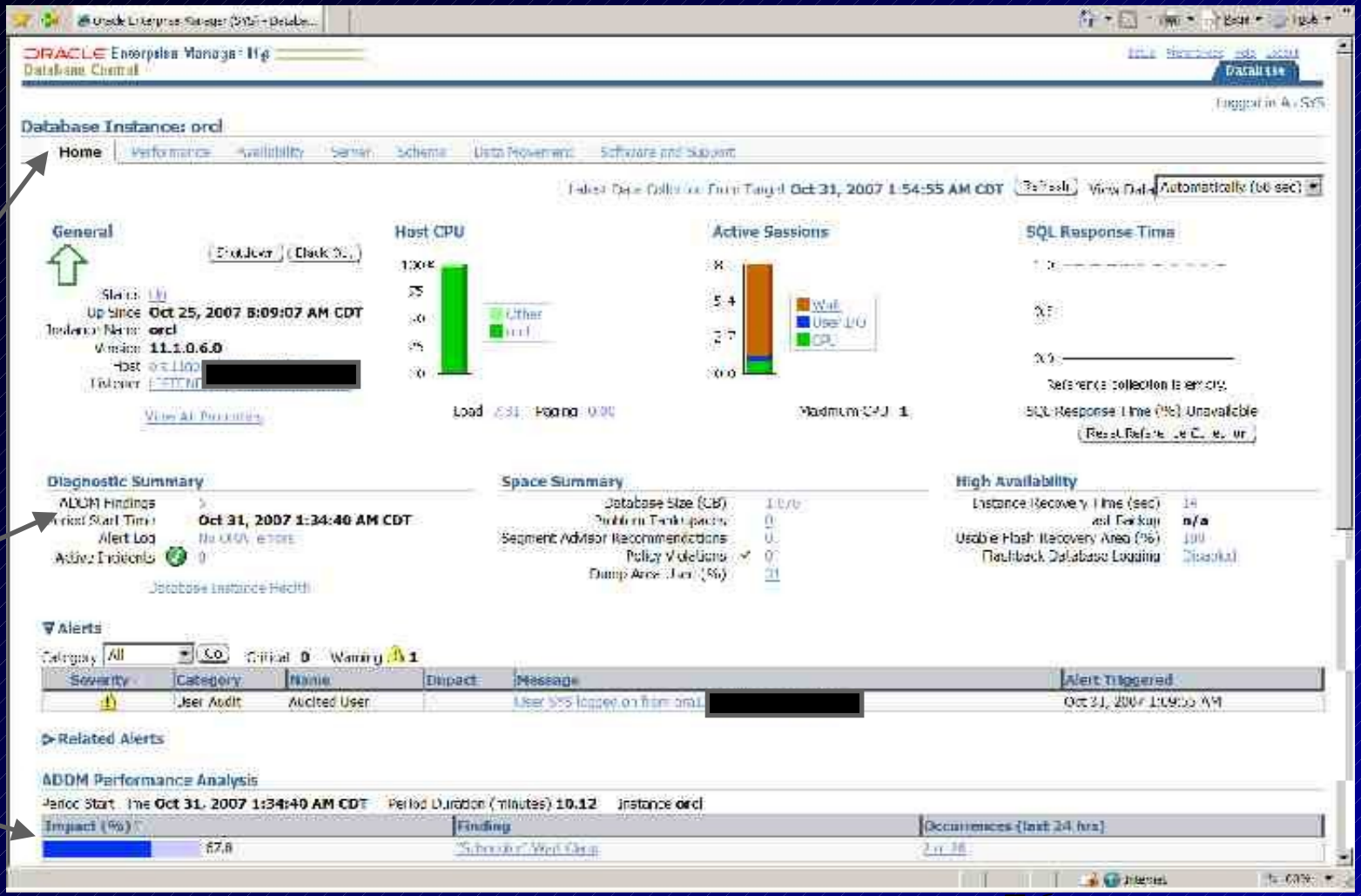
- Global ADDM so that Diagnostics are done across the entire cluster
- Emergency ADDM for use when database is hung
- On any granularity
  - Database Cluster
  - Database Instance
  - Specific Target (such as host, ASM...etc.)
- Over a specified time NOT tied to a pair of snapshots

ADDM Briefly

Specific Database Instance

We have 5 ADDM Findings

Check them Here

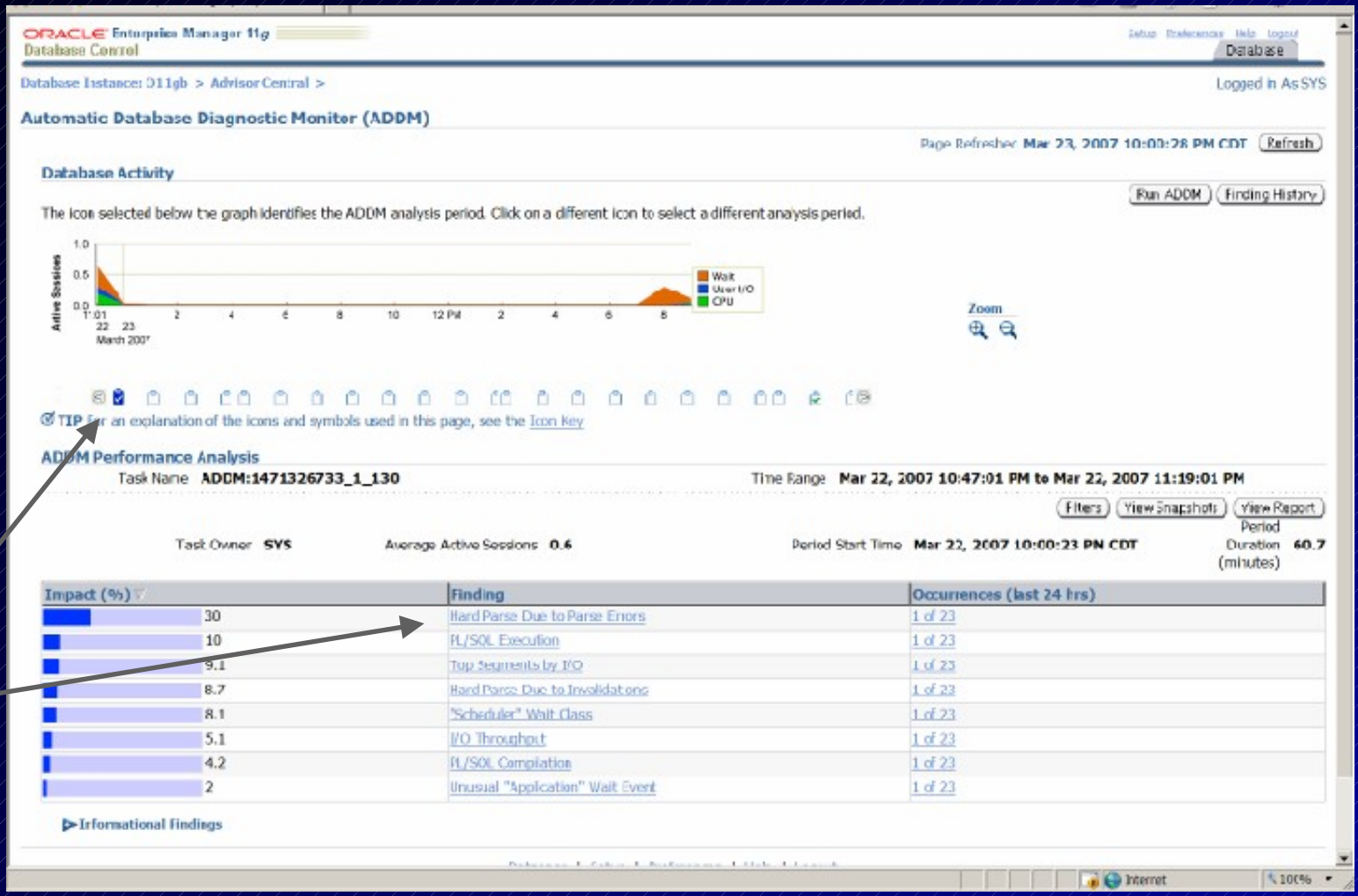


ADDM Briefly

Top ADDM Findings

Click a Single Timeframe

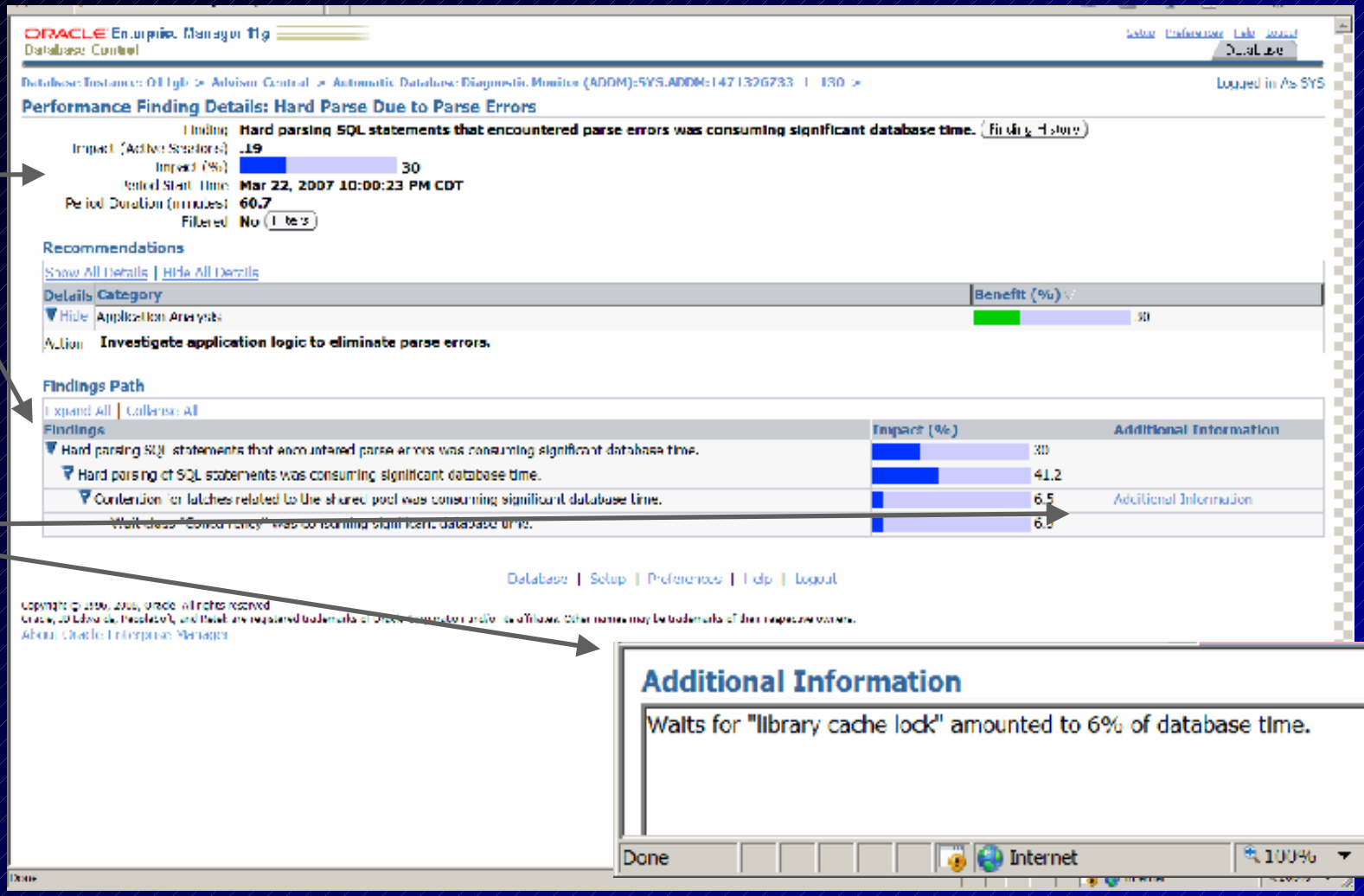
Let's Check the Hard Parse Issue



ADDM Briefly

Detailed Info & Findings

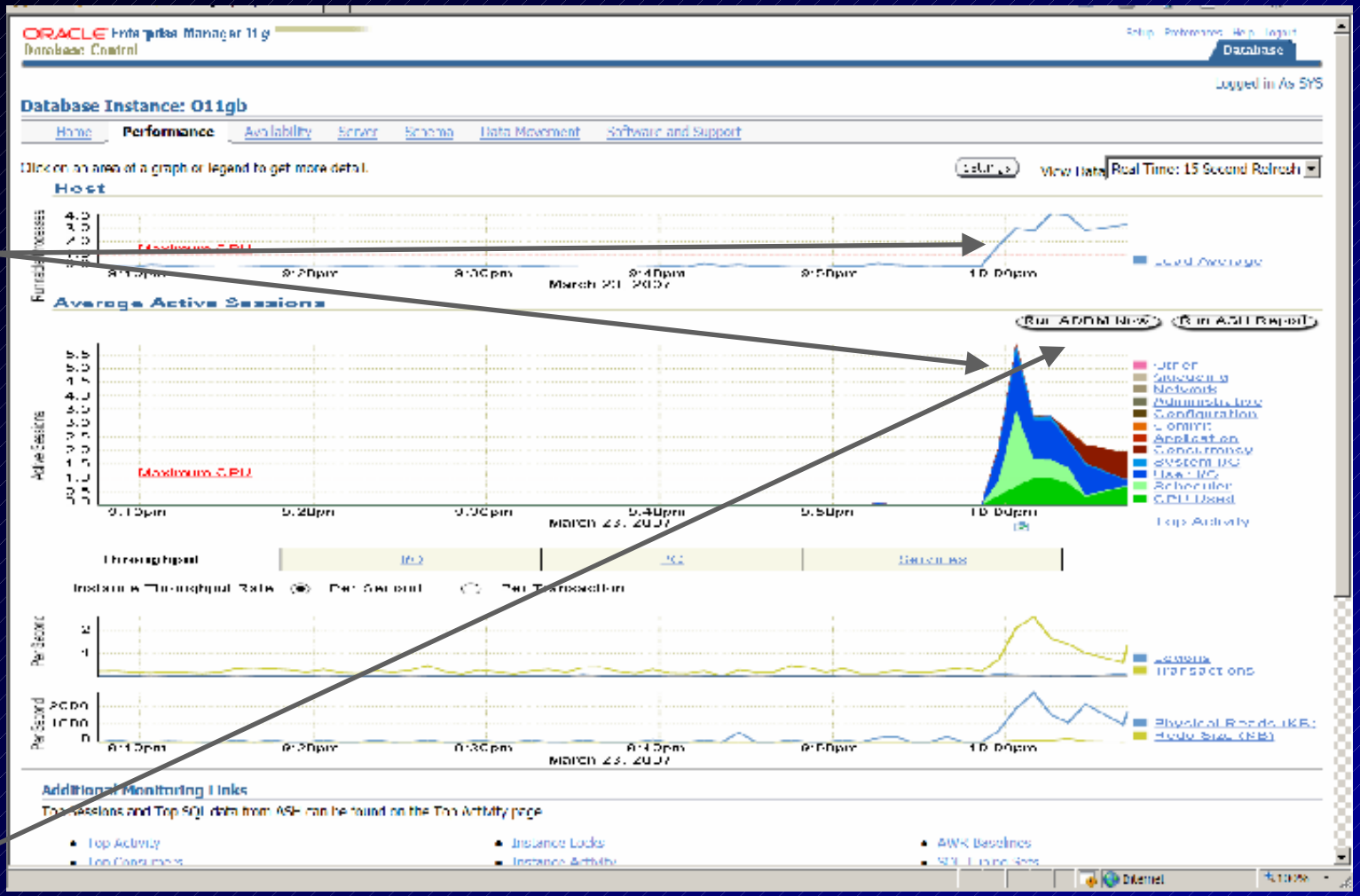
Add'l Info



ADDM - Run NOW!

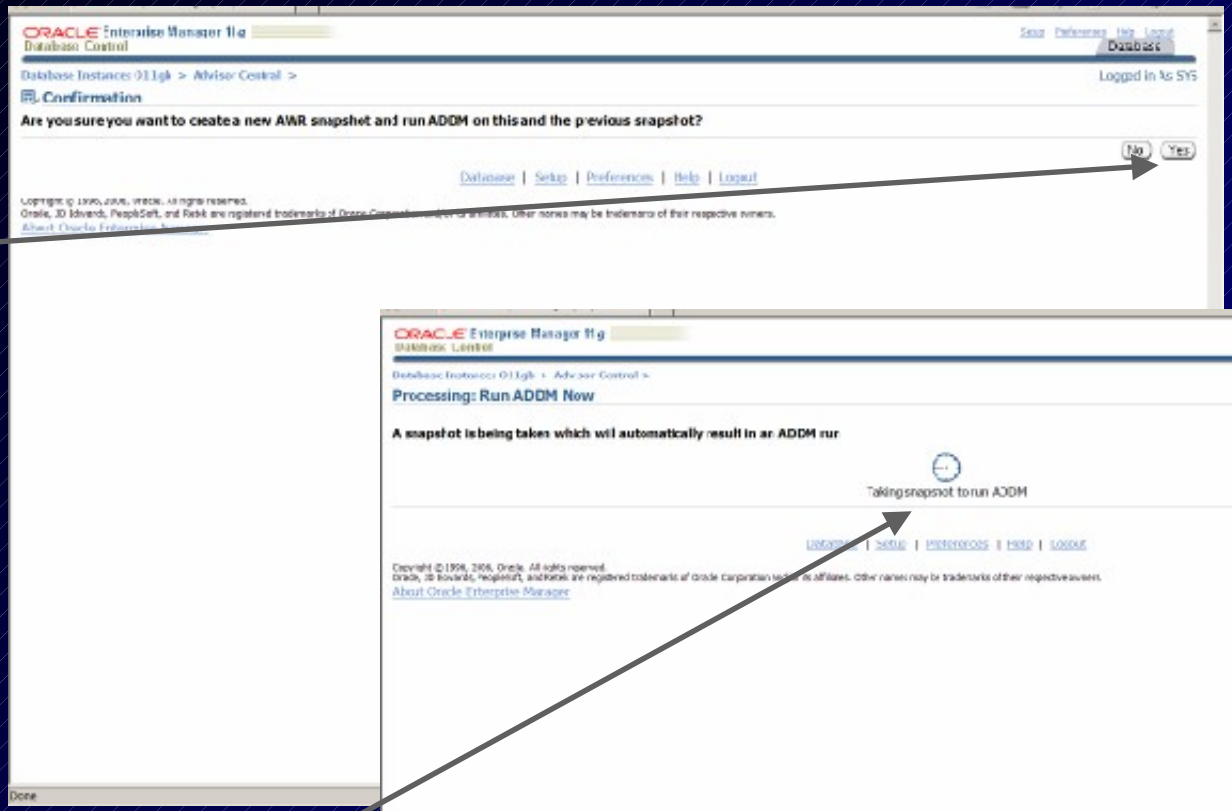
A Big Problem Occurs

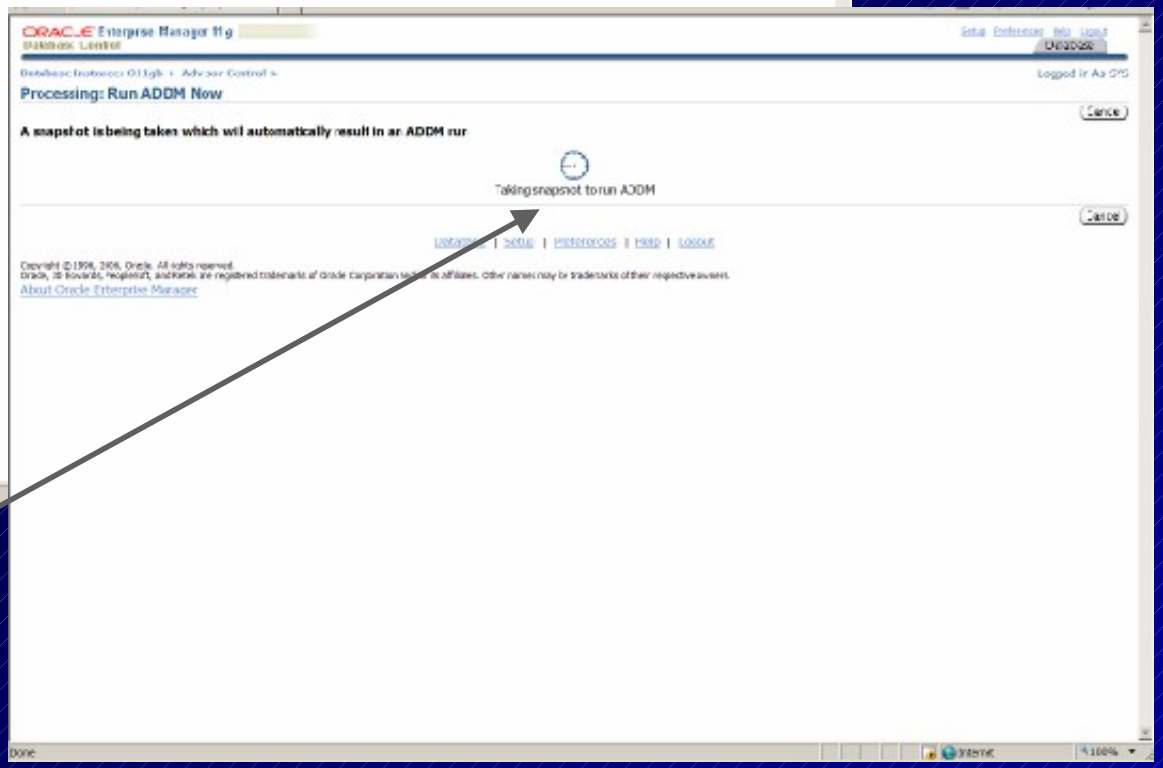
Run ADDM NOW!



ADDM - Run NOW!

Are you Sure?



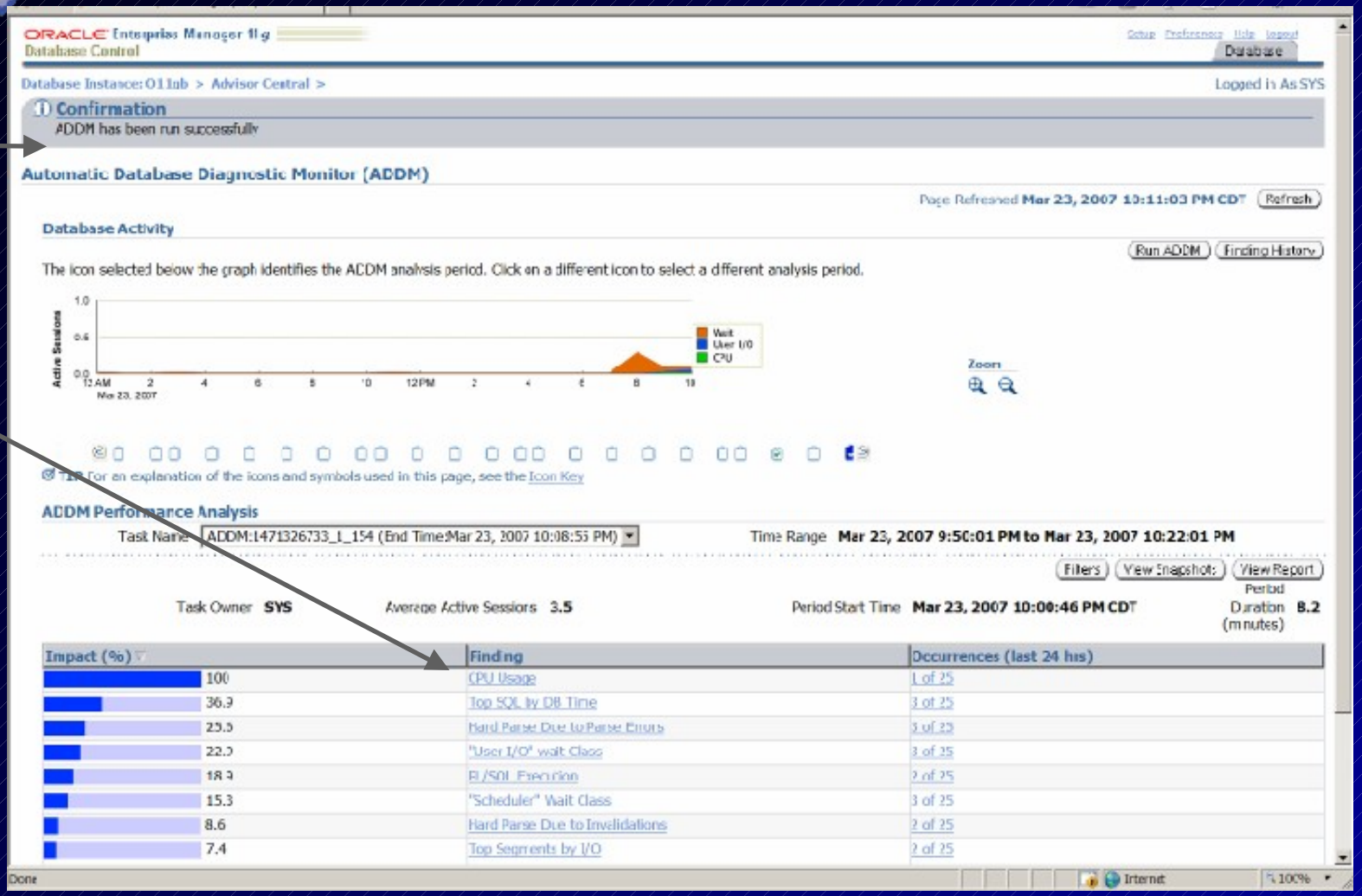


Running

ADDM – Run NOW!

Done.

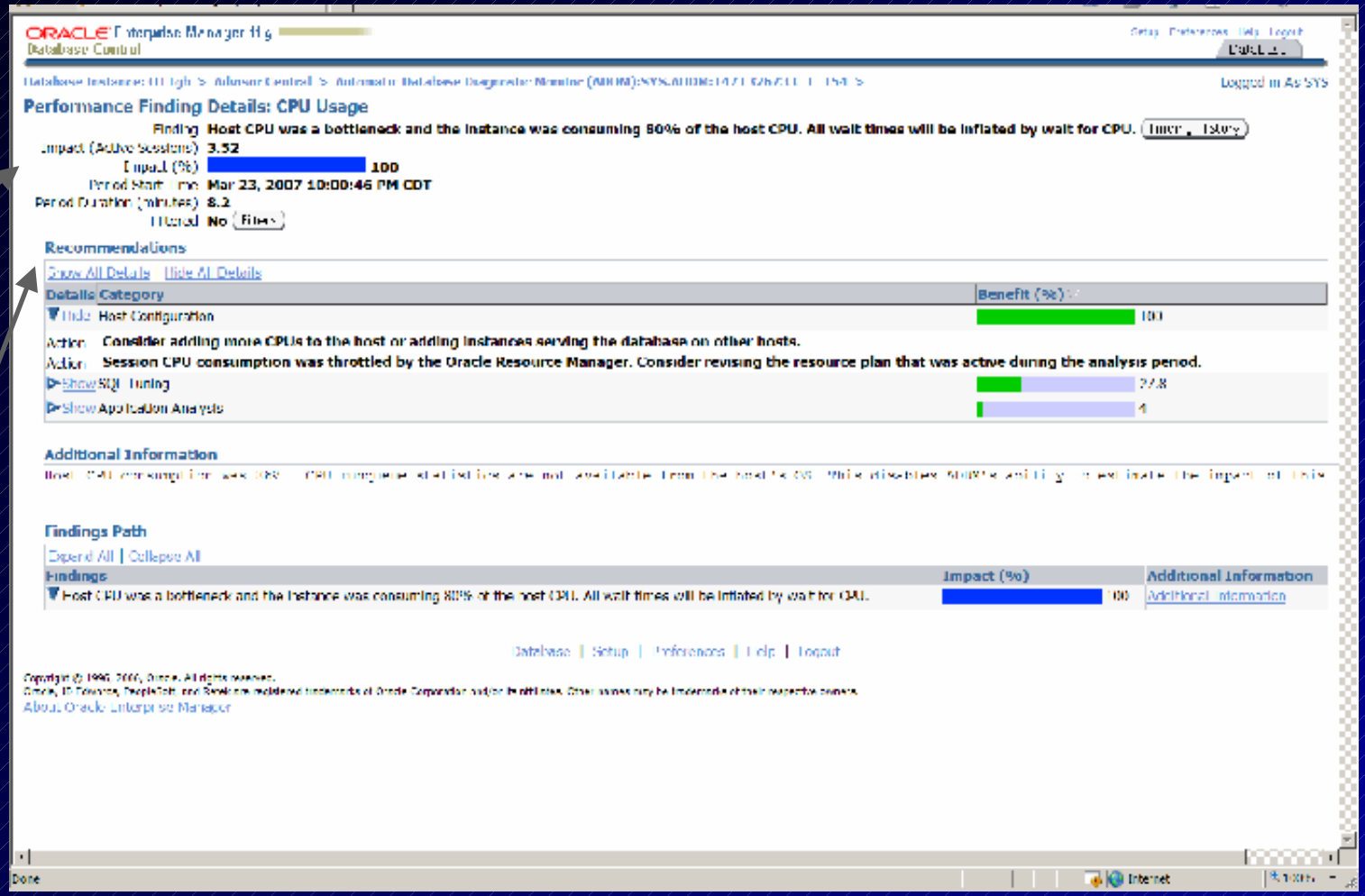
CPU Issue



ADDM – Run NOW!

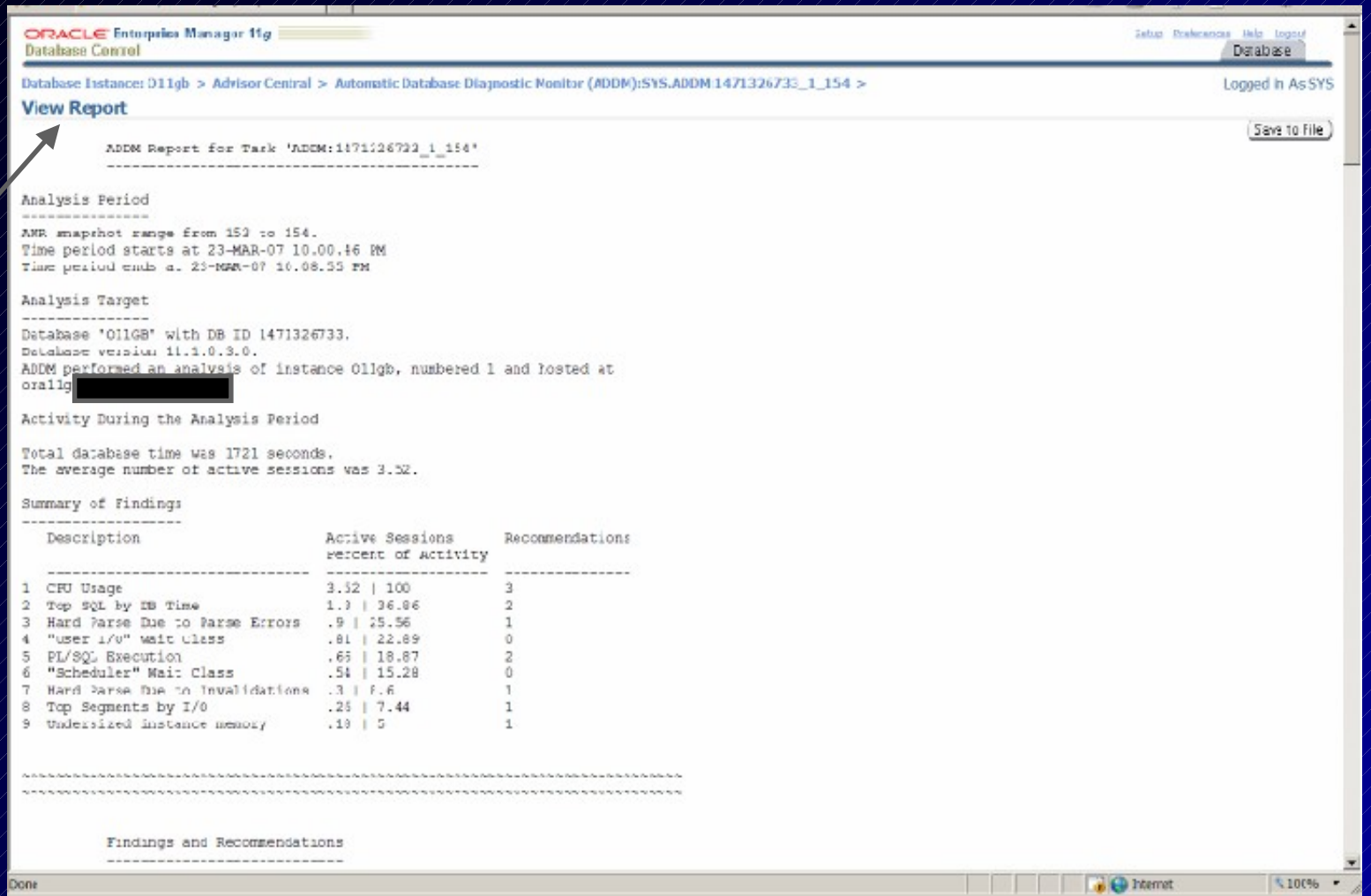
Detail on CPU Issue?

Suggested Fixes

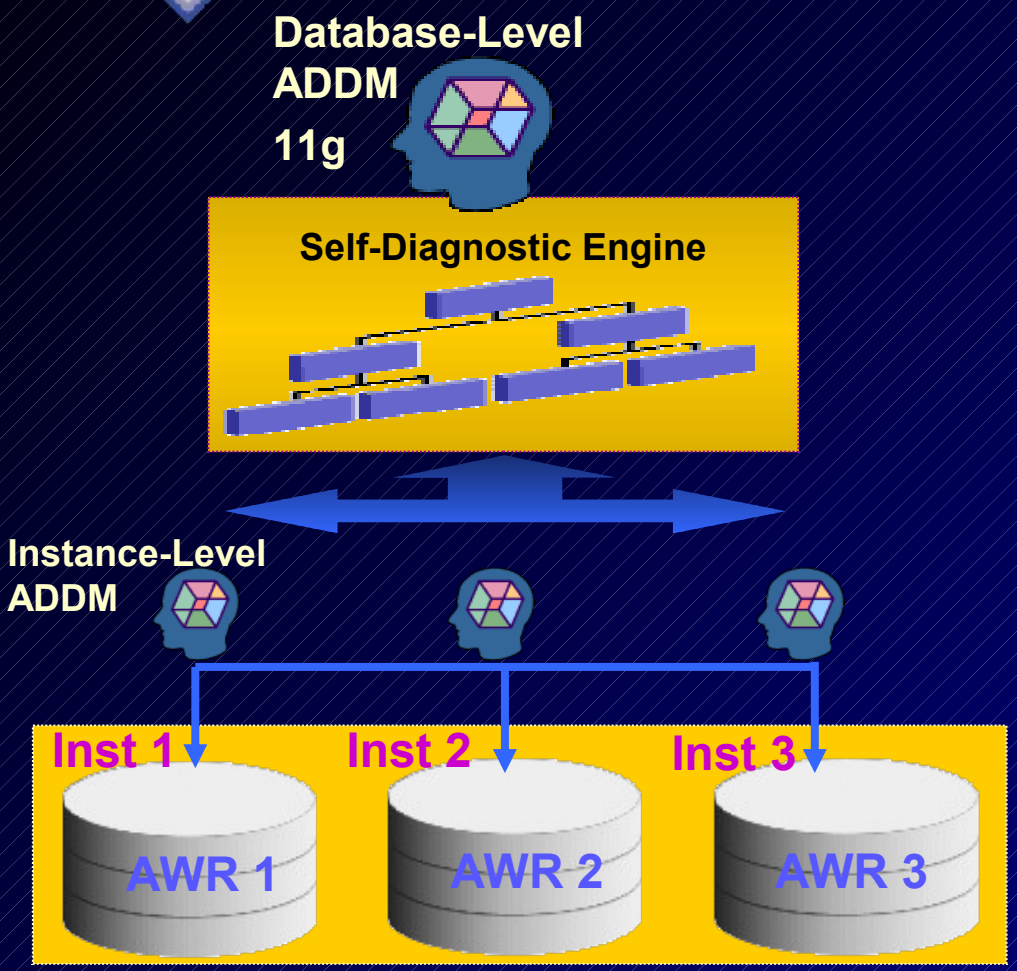


ADDM – Run NOW!

View The Report



ADDM for RAC

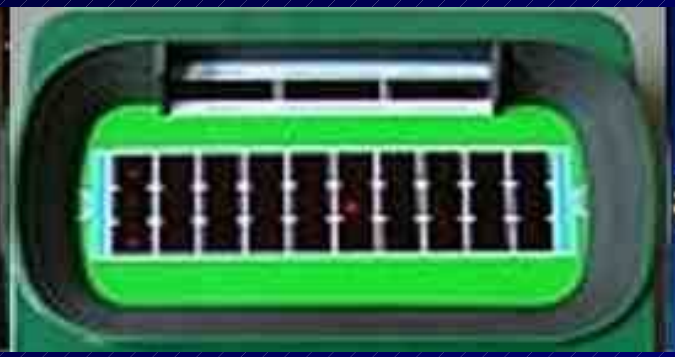


- Performance expert in a box
  - Now RAC specialist too!
- Identifies the most "Globally Significant" performance issues for the entire RAC database
- Database-wide and instance-level analysis
- Database-wide analysis of:
  - Global cache interconnect issues
  - Lock manager congestion issues
  - Global resource contention, e.g. IO bandwidth, hot blocks
  - Globally high-load SQL
  - Skew in instance response times
- Allows drill down to instances
- Runs proactively every hour when taking AWR Snapshots (default)

ADDM Considerations:

- CPU Bottlenecks
- Undersized Memory Structures – SGA / PGA
- I/O Capacity Issues
- High Load SQL statements
- High Load PL/SQL
- RAC specific issues – Global hot block/interconnect
- Application issues such as parsing, locks...etc.
- Concurrency (buffer busy) or hot object issues
- Configuration issues – Redo, Archive, Checkpoint.

SQL Tuning Advisors & SQL Plan Management (SPM)



SQL Plan Management

- SQL Plan Management is a mechanism that records/evaluates execution plan of SQL statements (good & bad) over time and builds SQL Plan baselines (replaces stored outlines) of existing plans known to be efficient.
- Events that cause the need for SQL Plan baselines:
  - New version of Oracle (New optimizer version – Use capture replay to test effect)
  - Changes to optimizer statistics or data changes
  - Schema, application or metadata changes (use SQL Advisor to get suggestions)
  - System settings changes (Use SQL Replay to find what works)
  - SQL Profile (statistics – data skews & correlated columns) creation
- Stored outlines are deprecated (discouraged) in Oracle Database 11*g*. Oracle highly recommends migrating existing stored outlines to SQL plan baselines. A **SQL Profile contains additional STATISTICS** for this SQL statement for the query optimizer to generate a better execution plan. **An outline/baseline contains HINTS** for this SQL statement for query optimizer to generate a better execution plan.

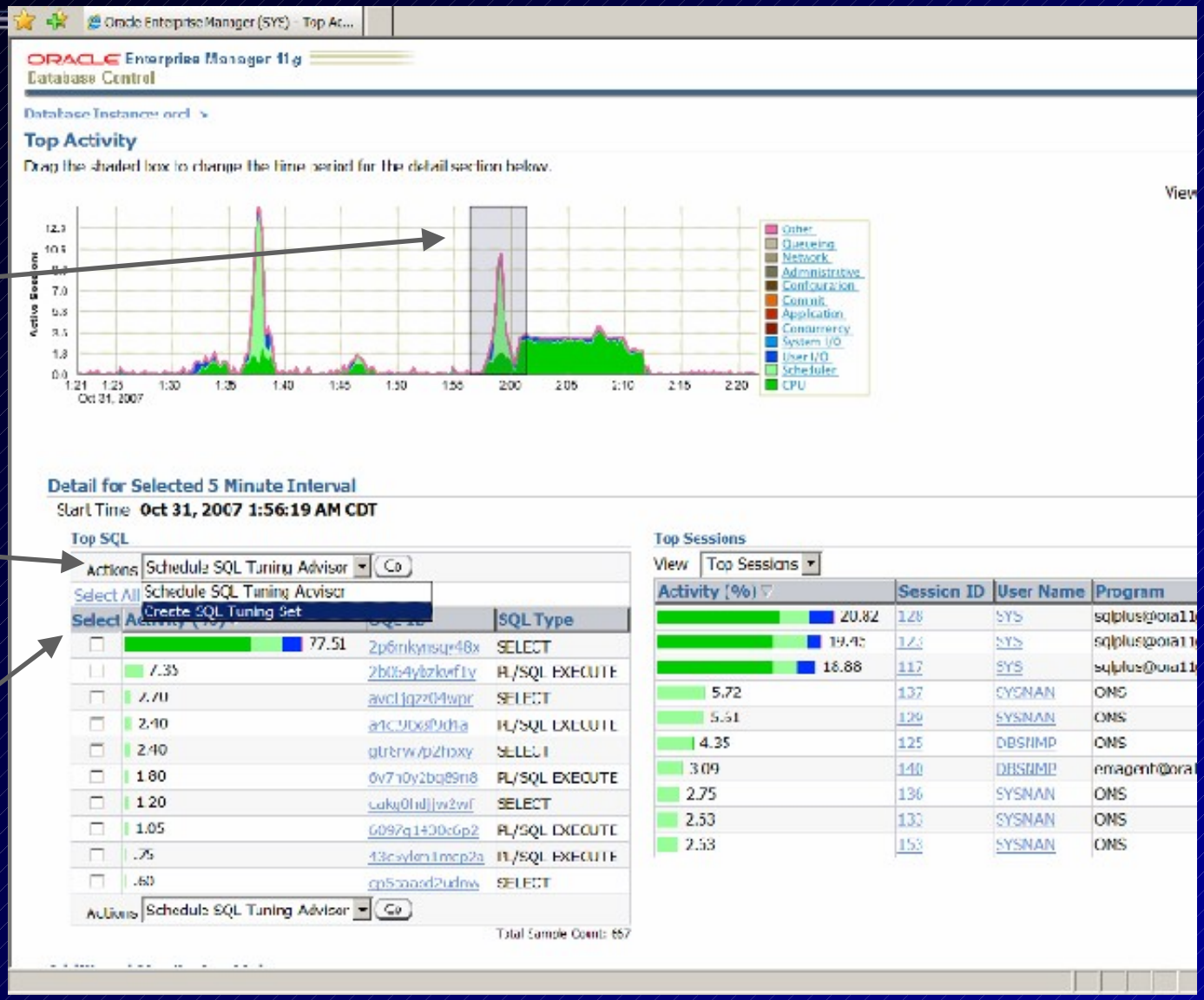
SQL Plan Management

- **SQL Profile stores STATISTICS** for a SQL statement for the query optimizer to generate a better execution plan.
- **A Stored Outline/SQL Plan Baseline contains HINTS** for this SQL statement for query optimizer to generate a better execution plan.
- A SQL Plan Baseline should evolve with changes in the system to analyze good/bad plans over time.
- View these in DBA_PLAN_BASELINES
- You can also export a SQL Tuning Set and import it to new system. Capture baselines for Tuning Set with DBMS_SPM (see later slide on entire syntax). Can also use a pack/unpack function to pack/unpack all plans in a system for transporting.

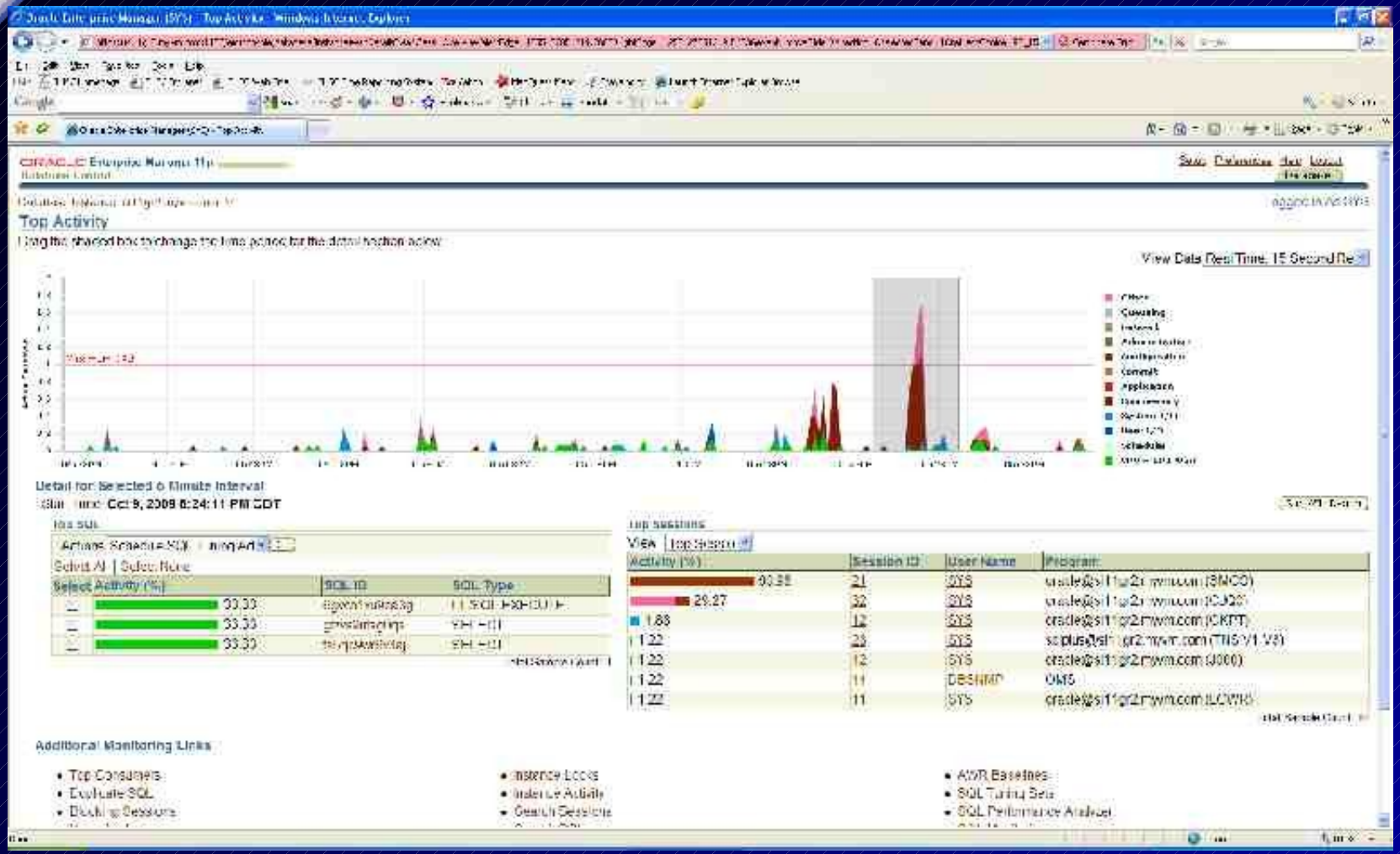
SQL Plan Management
Create a SQL Tuning Set

Tuning Issue

Create a Tuning Set from Top 10 SQL



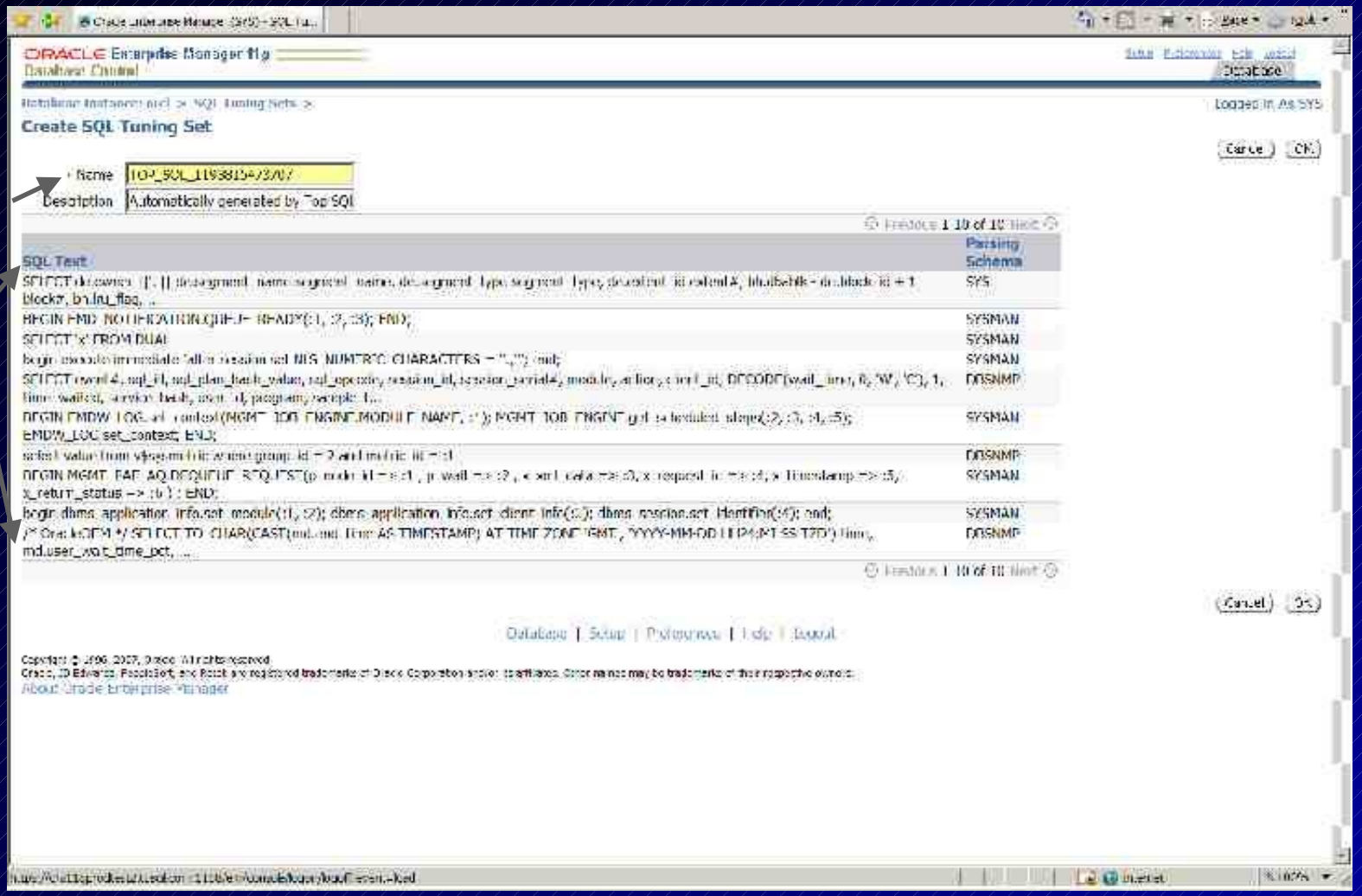
Top Activity – 11gR2 (same look)



SQL Plan Management
Create a SQL Tuning Set

Tuning Set Name

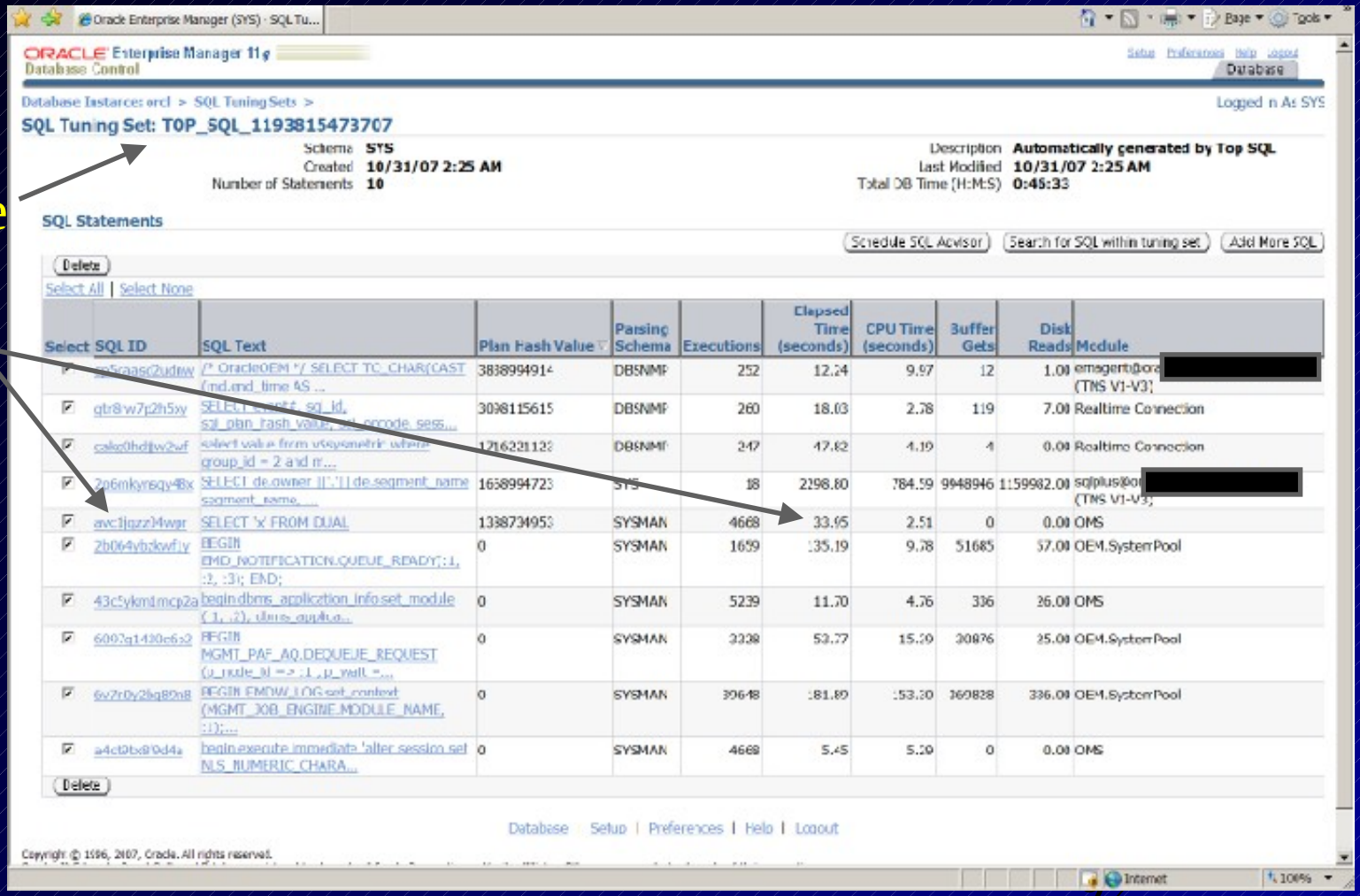
Queries



SQL Plan Management
Viewing a SQL Tuning Set

Tuning Set Name

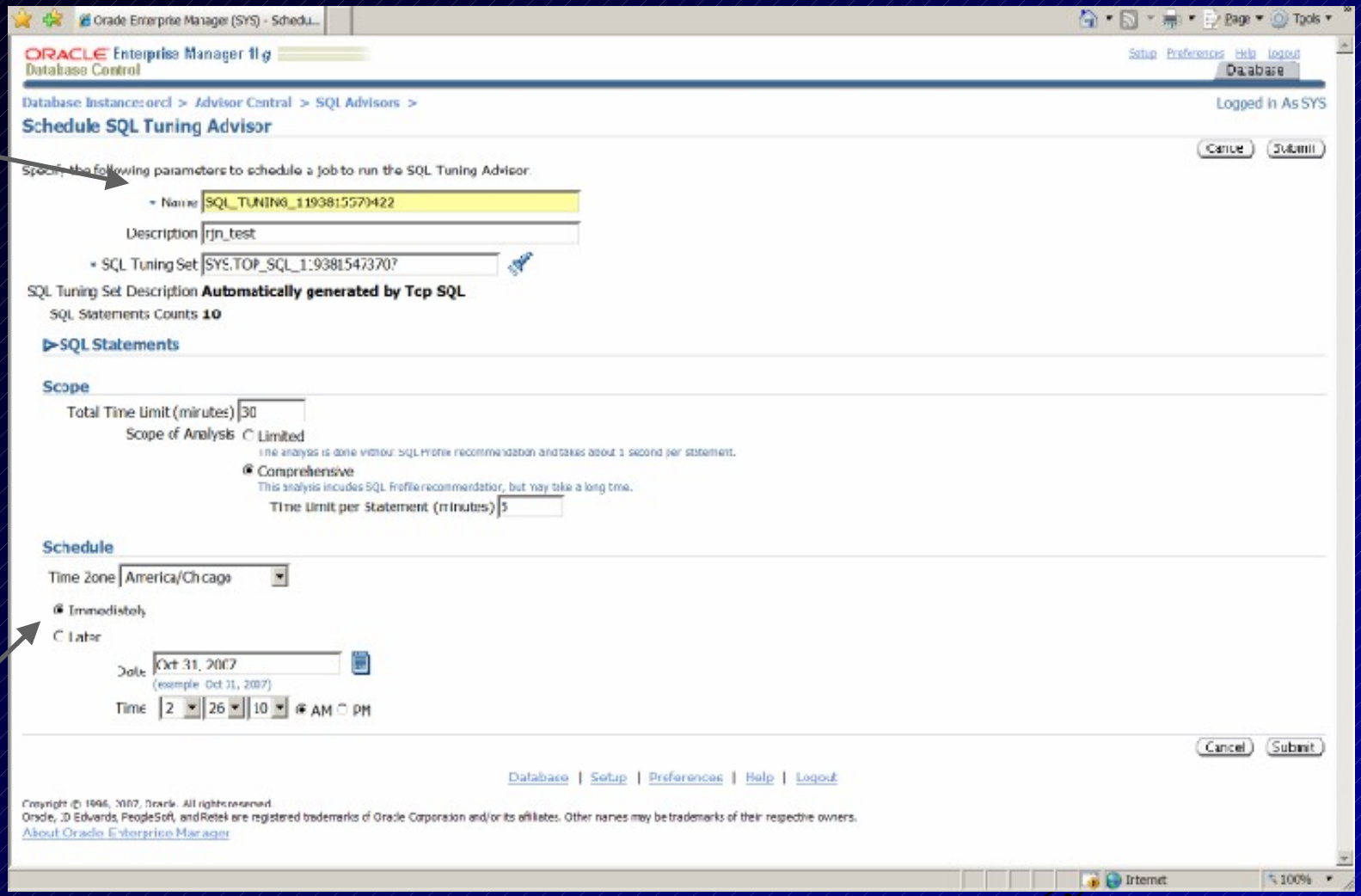
Queries & Stats



SQL Plan Management
Create a SQL Tuning Set

Run the Tuning Advisor on this SQL Tuning Set (STS)

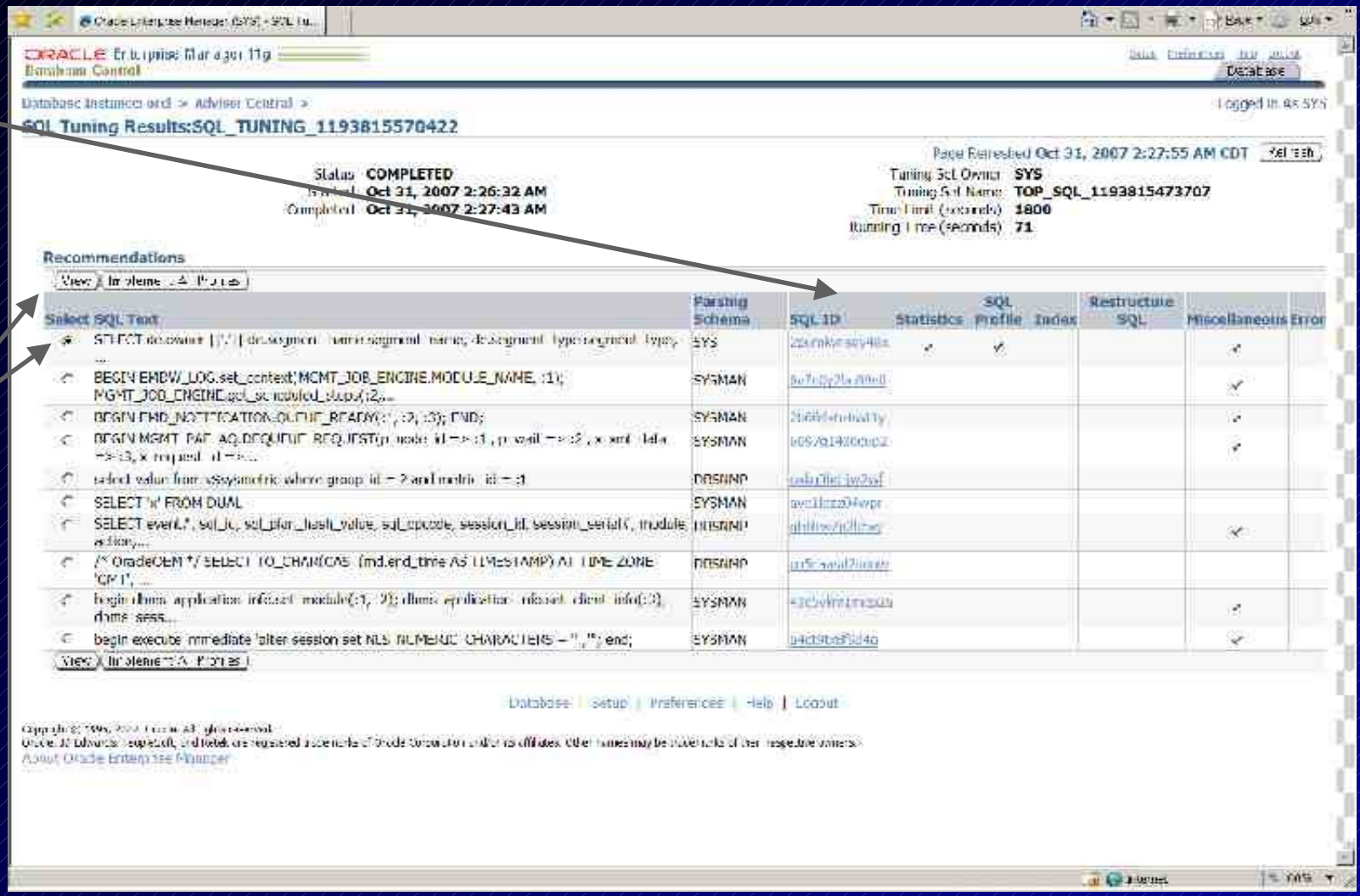
Run it NOW



SQL Plan Management
Create a SQL Tuning Set

Results

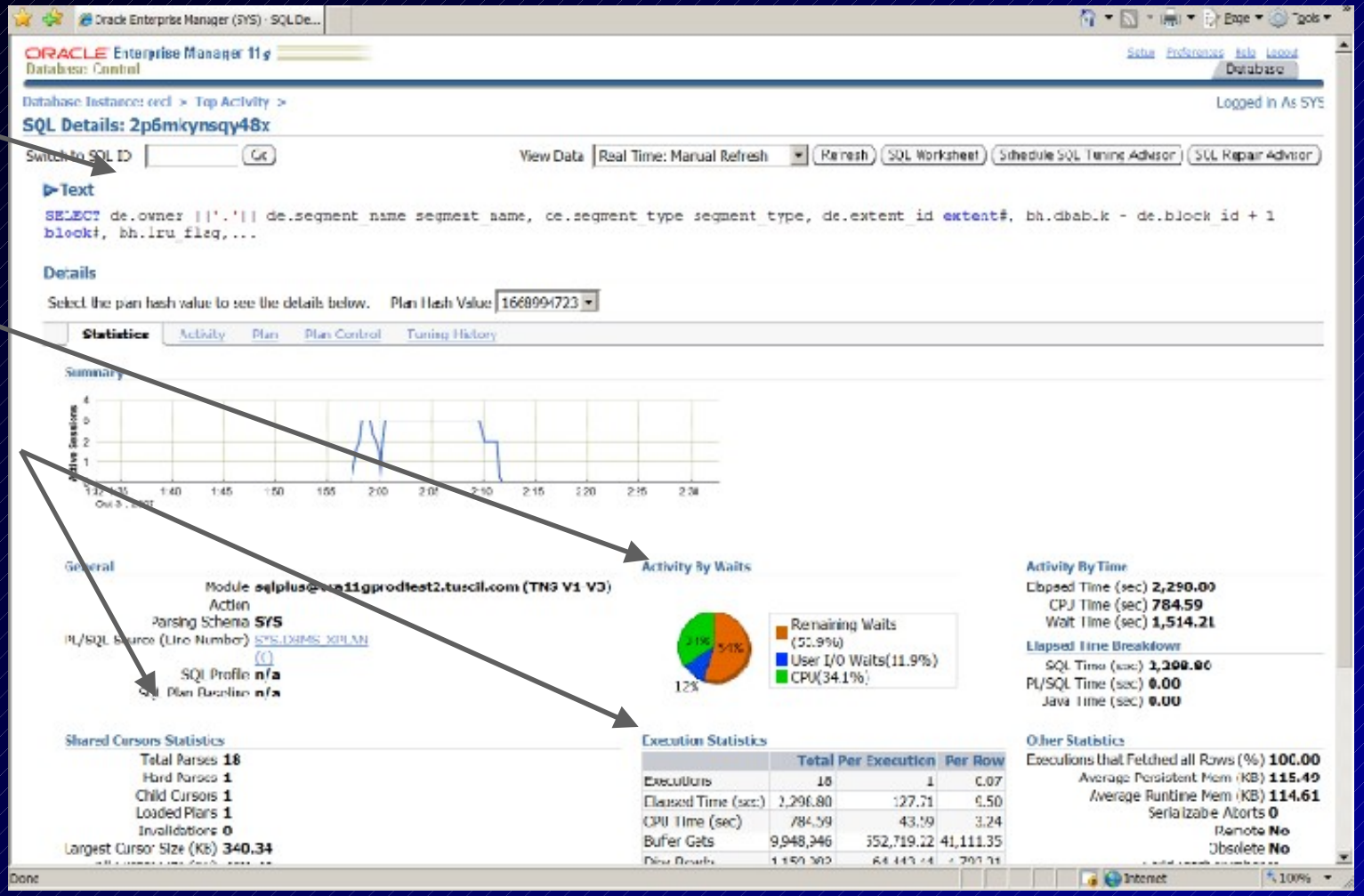
Select
One query
And click
View



SQL Plan Management
Click on any SQL ID

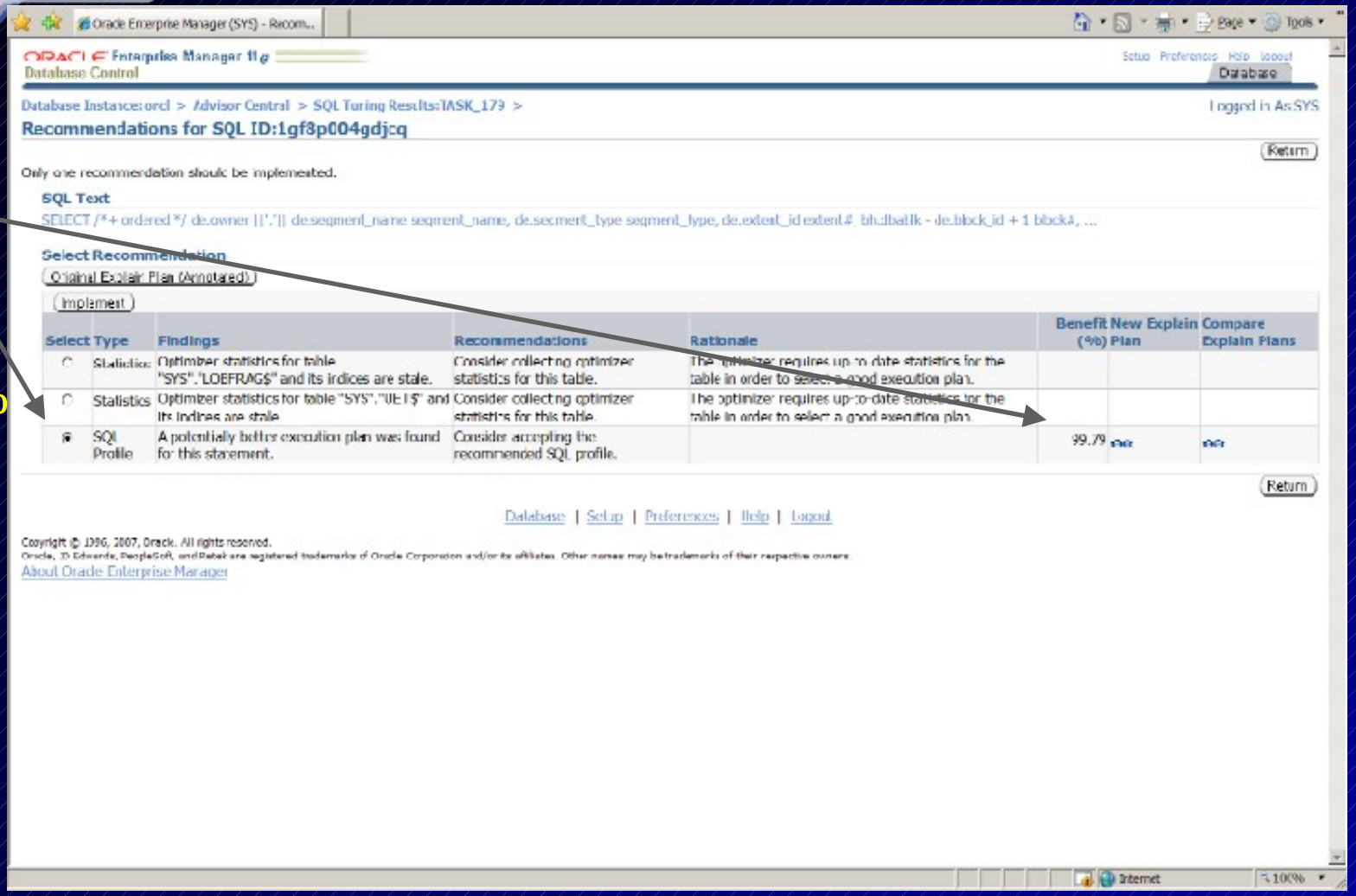
SQL Text

Waits & Statistics



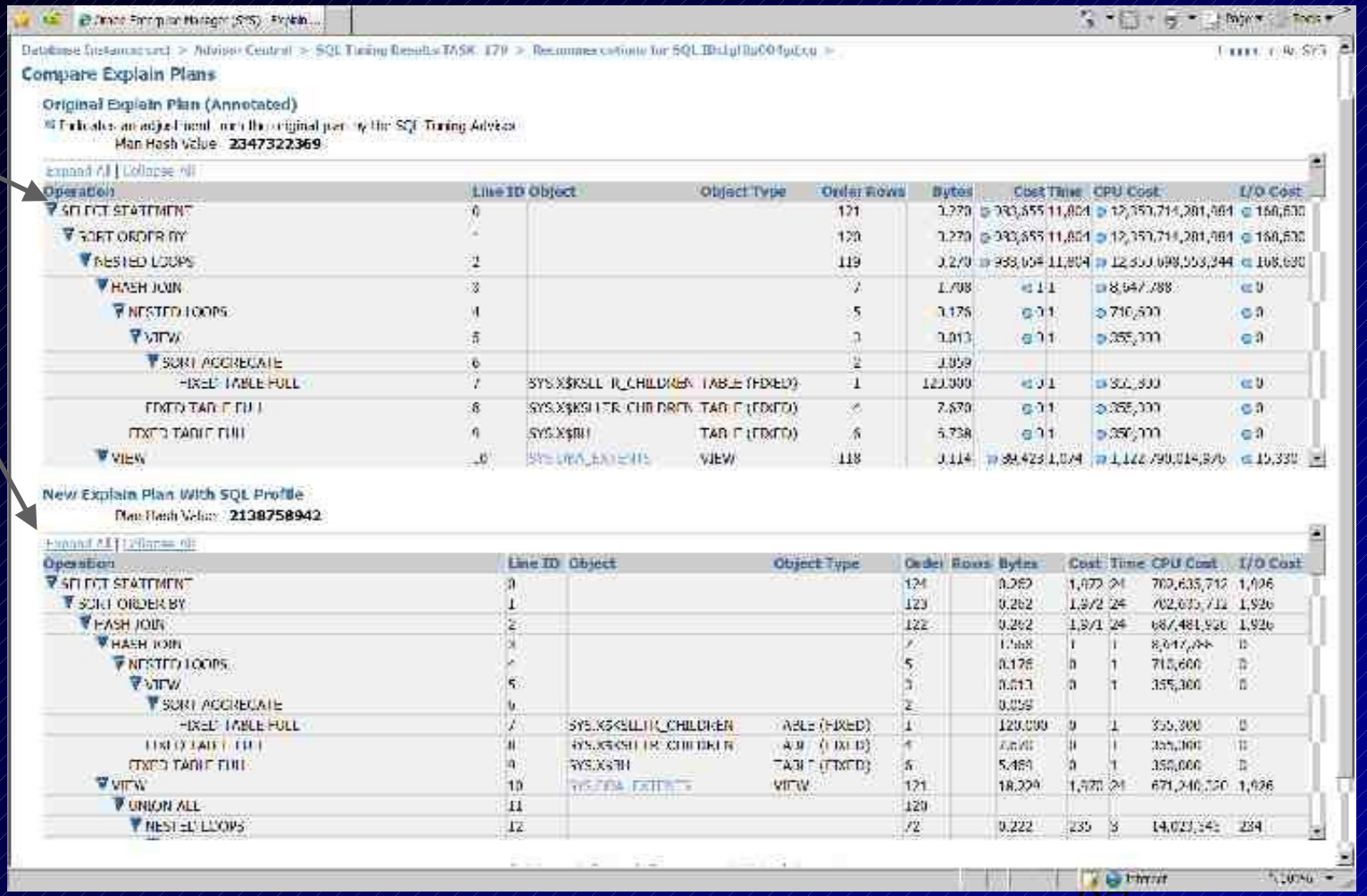
SQL Plan Management
Create a SQL Tuning Set

SQL Profile Will Help 99%



SQL Plan Management
Create a SQL Tuning Set

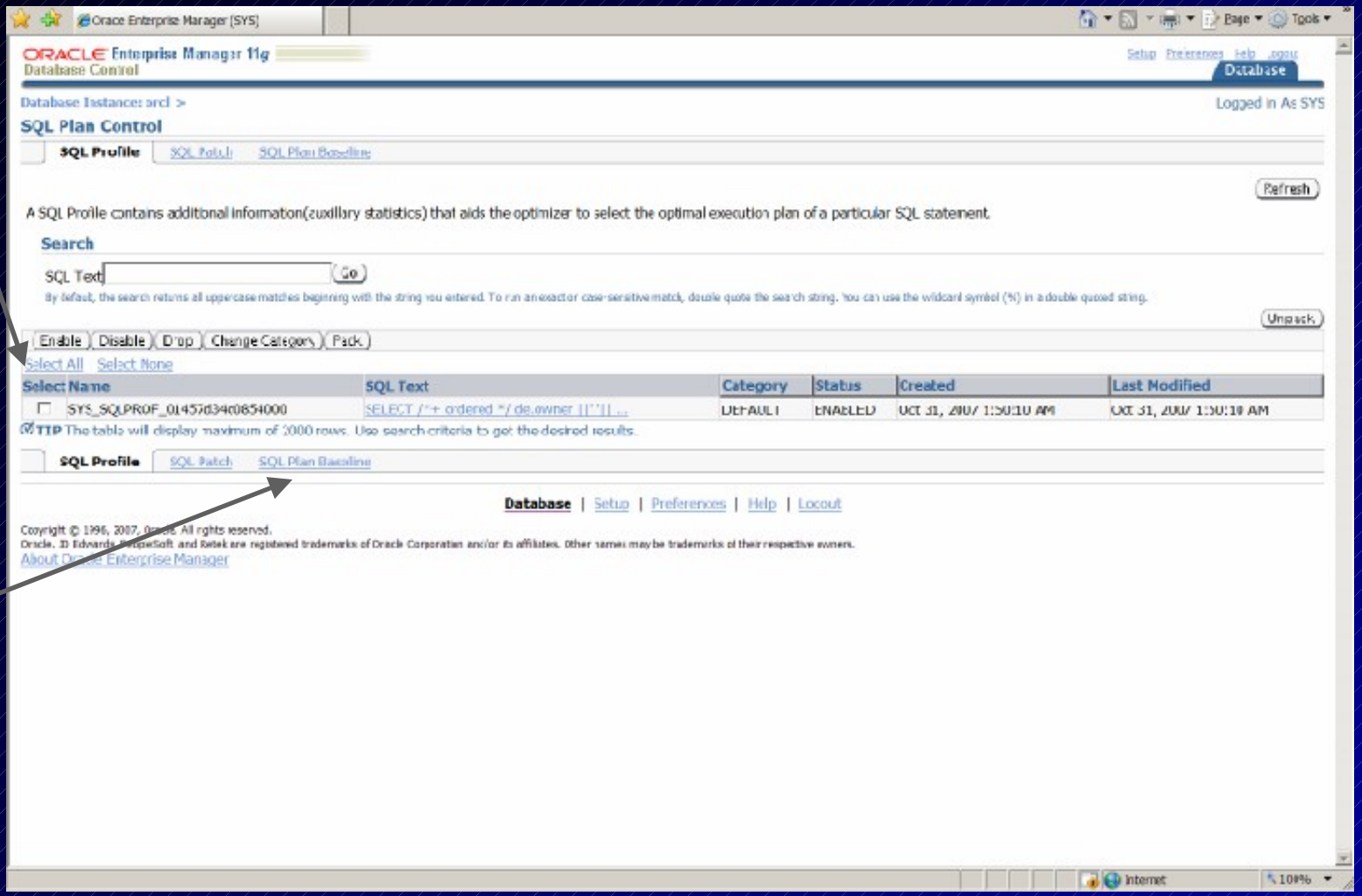
Compare Before & After



SQL Plan Control
SQL Profiles stored in the system

SQL Profiles

SQL Plan Baselines



SQL Plan Management
Capturing Baselines

- Capturing baselines (migrate stored outlines - hints)
- Plan history is only tracked for a SQL statement that executes more than once (no ad-hoc queries)
- Automatic Plan Capture:
  - OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES
  - (set to TRUE – the default is FALSE)
- Enable the use of SQL Plan Baselines (could be session level of a tuning set & without the capture):
  - OPTIMIZER_USE_SQL_PLAN_BASELINES
  - (set to TRUE – the default is TRUE)

SQL Plan Management
Capturing Baselines

```
DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE (
        sql_id IN VARCHAR2,
        plan_hash_value IN NUMBER := NULL,
        sql_text IN CLOB,
        fixed IN VARCHAR2 := 'NO',
        enabled IN VARCHAR2 := 'YES')
 RETURN PLS_INTEGER;
```

- **Load one or more plans present in the cursor cache for a SQL statement.** You can also do this using the plan_handle and sql_text.

SQL Plan Management
Capturing Baselines

```
DBMS_SPM.LOAD_PLANS_FROM_SQLSET (
        sqlset_name IN VARCHAR2,
        sqlset_owner IN VARCHAR2 := NULL,
        basic_filter IN VARCHAR2 := NULL,
        fixed IN VARCHAR2 := 'NO',
        enabled IN VARCHAR2 := 'YES'
        commit_rows IN NUMBER := 1000)
RETURN PLS_INTEGER;
```

- **Manually load plans stored in SQL Tuning Sets (STS) into plan baselines.**
- Note that plan history is only tracked for a SQL statement that executes more than once (no ad-hoc queries)

SQL Plan Management
Capturing Baselines

- Set the retention to 100 weeks to retain **unused** plans before they are purged (53 is the default). Shown for example, 100 is not recommended, 53 is better since it will include monthly/yearly runs.

```
begin
   dbms_spm.configure('plan_retention_weeks',100);
end;
/
```

PL/SQL procedure successfully completed.

- You can also purge individual plans with the purge_sql_plan_baseline function.
- You can also query dba_sql_plan_baselines as well as use DBMS_XPLAN.display_sql_plan_baseline to view stored plans.

SQL Performance Analyzer

SQL Performance Analyzer

- Measure and report on performance before and after a change! DBMS_SQLTUNE package.

Great for:

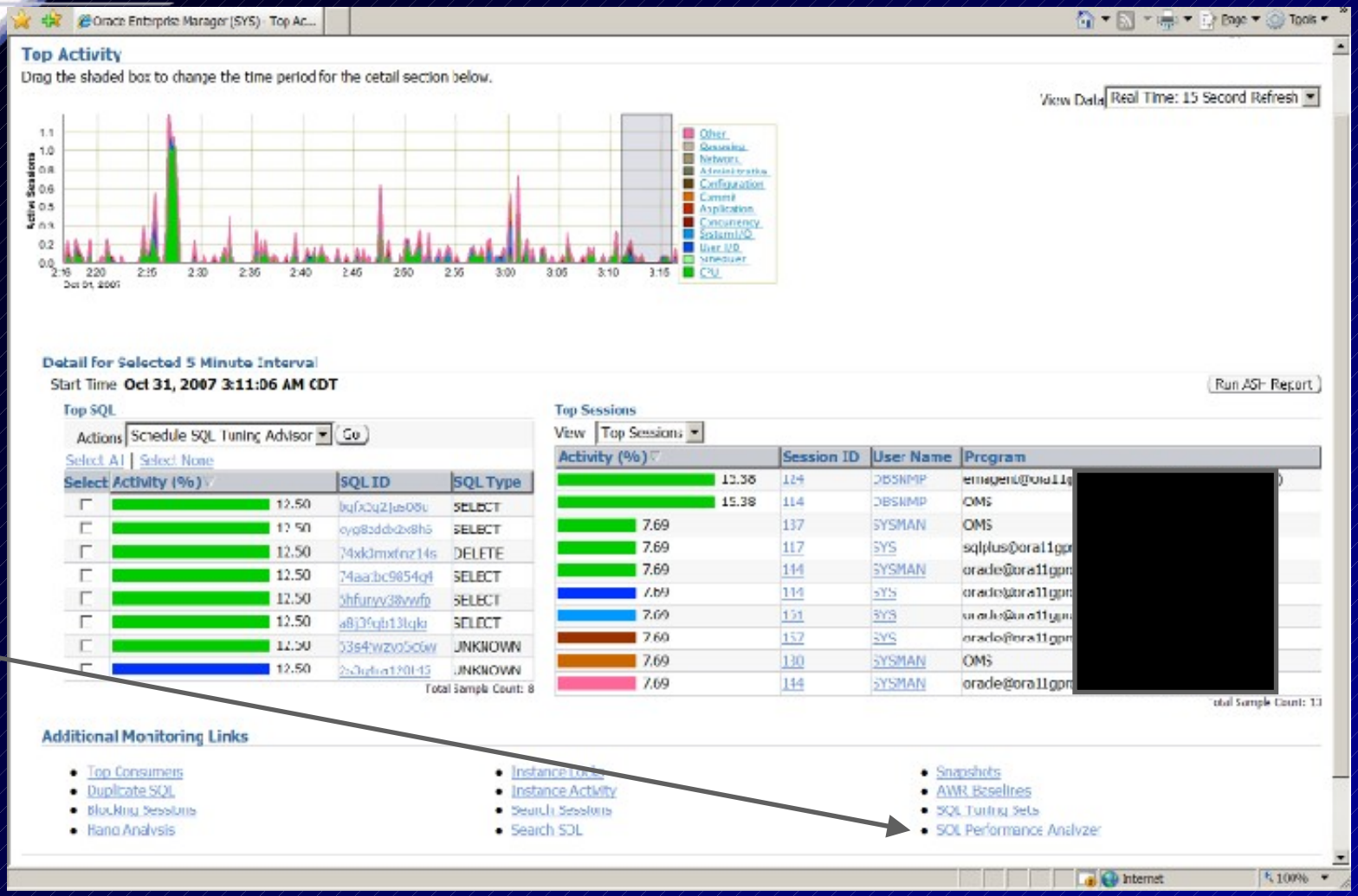
- Database Upgrades
- Application Upgrades
- Hardware Changes
- Database or Schema Changes
- Best for SQL Tuning – Especially Batches

SQL Performance Analyzer

Easy to run – SQL Focus (Test SGA settings):

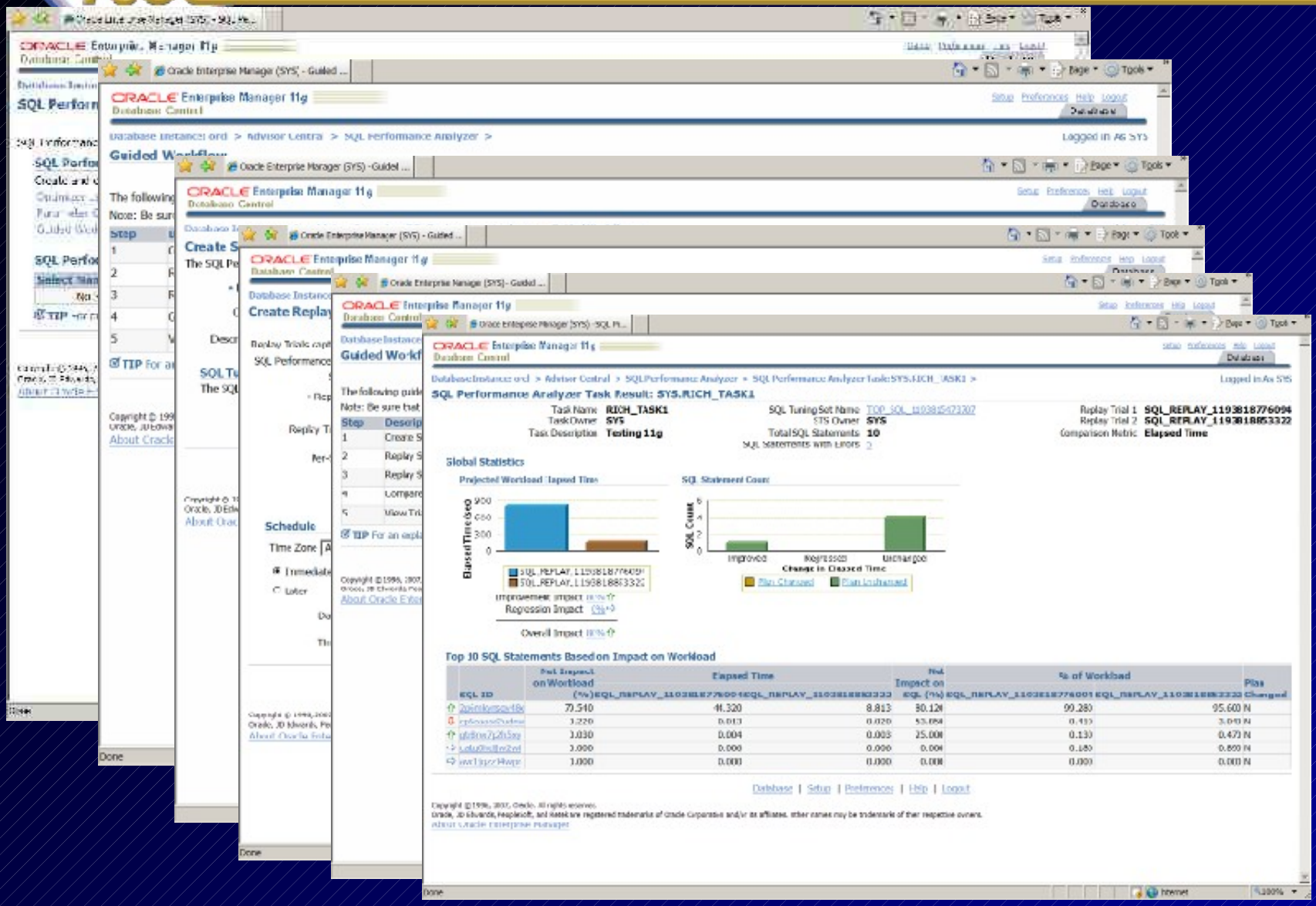
- Capture SQL
- Transport SQL
- Create a Replay Task
- Set up the environment to Test
- Make any changes to Test (such as SGA/Optimizer)
- Compare before and after performance
- Tune the problems!

SQL Performance Analyzer

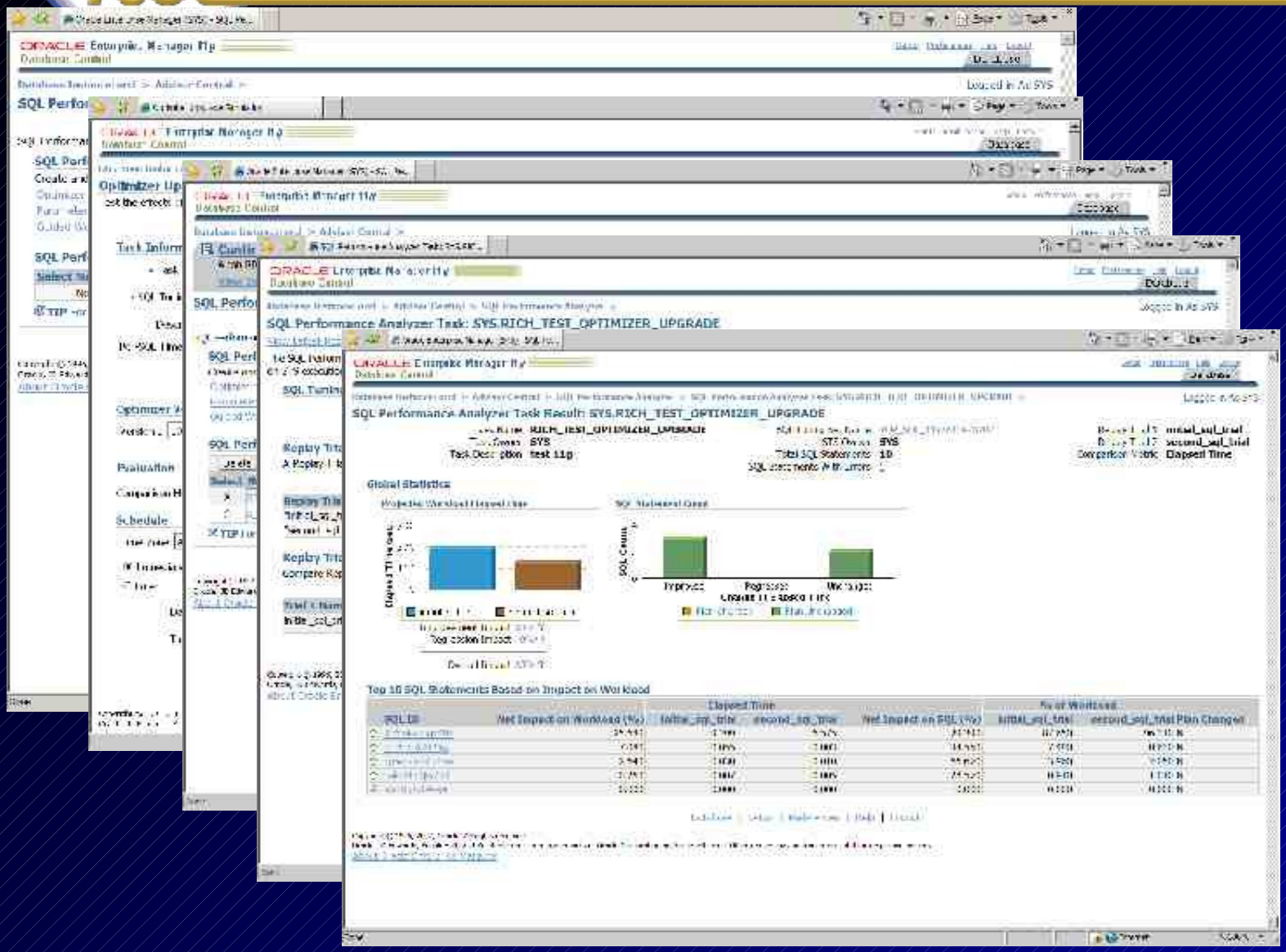


Click Here

SQL Performance Analyzer Guided Workflow

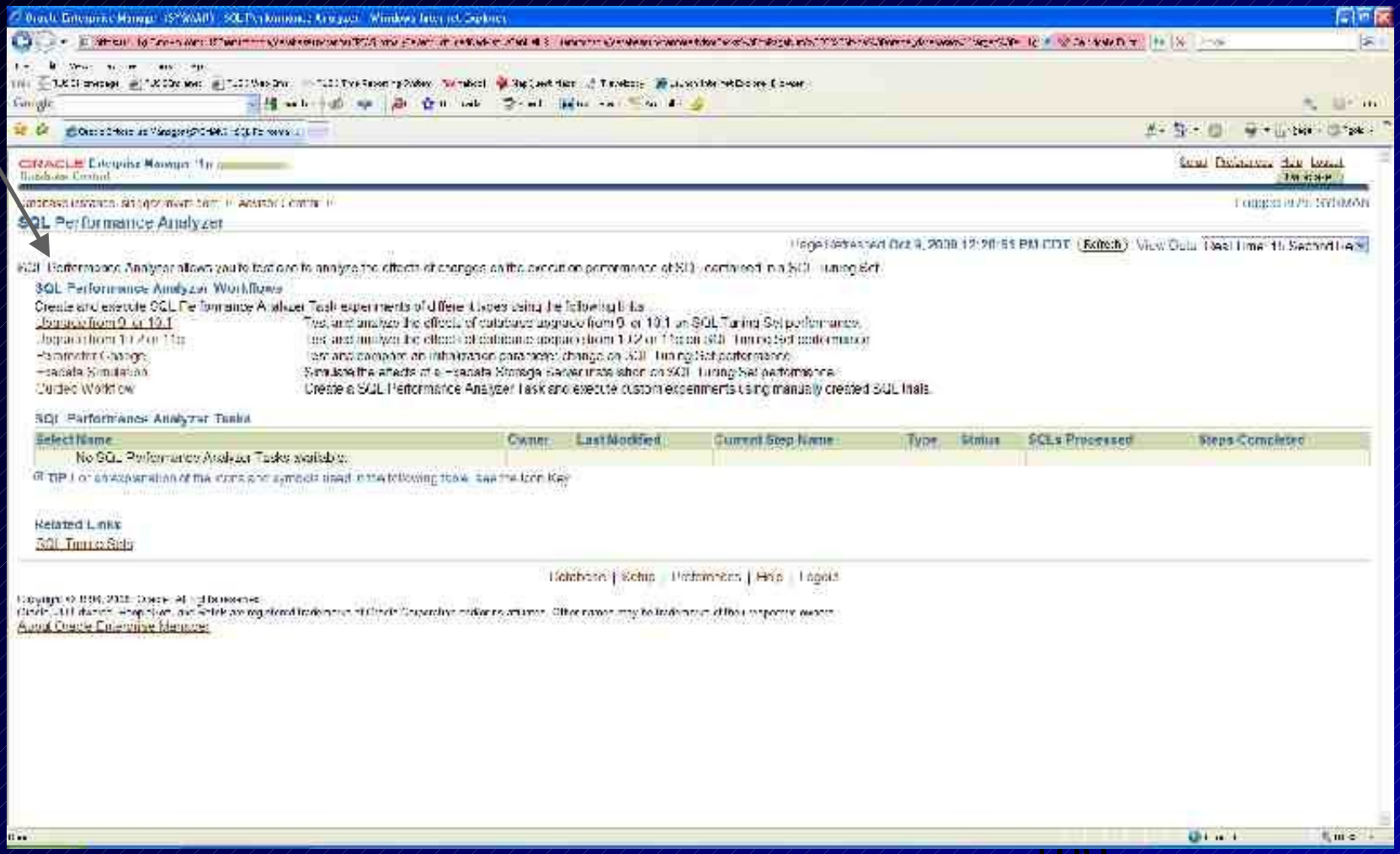


SQL Performance Analyzer Optimizer Upgrade (10g vs. 11g)



SQL Performance Analyzer 11gR2 - Options

Upgrade Options

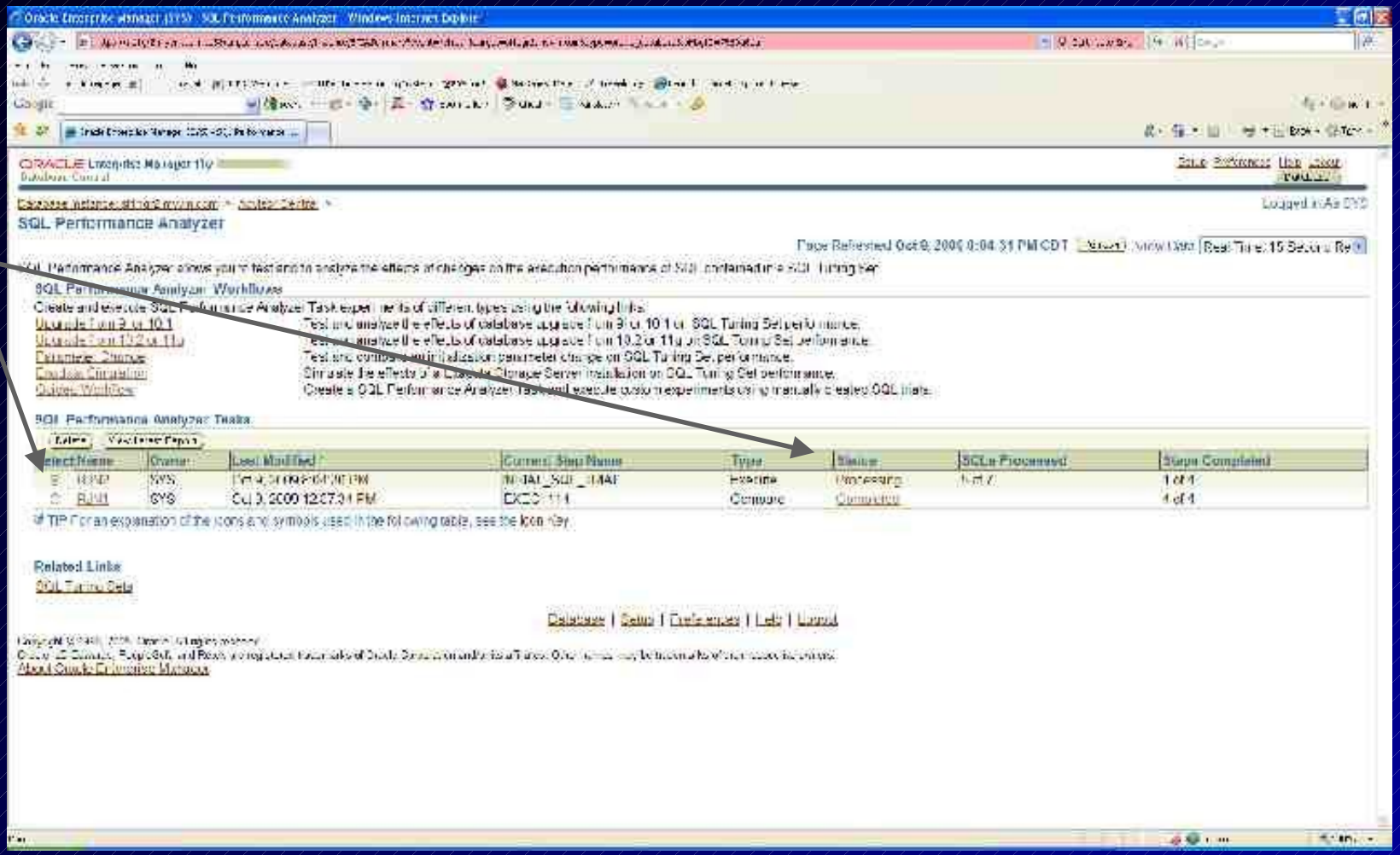


SQL Performance Analyzer 11gR2 – Exadata Simulation

Test a Tuning Set that I've used in the past

SQL Performance Analyzer 11gR2 – Exadata Simulation

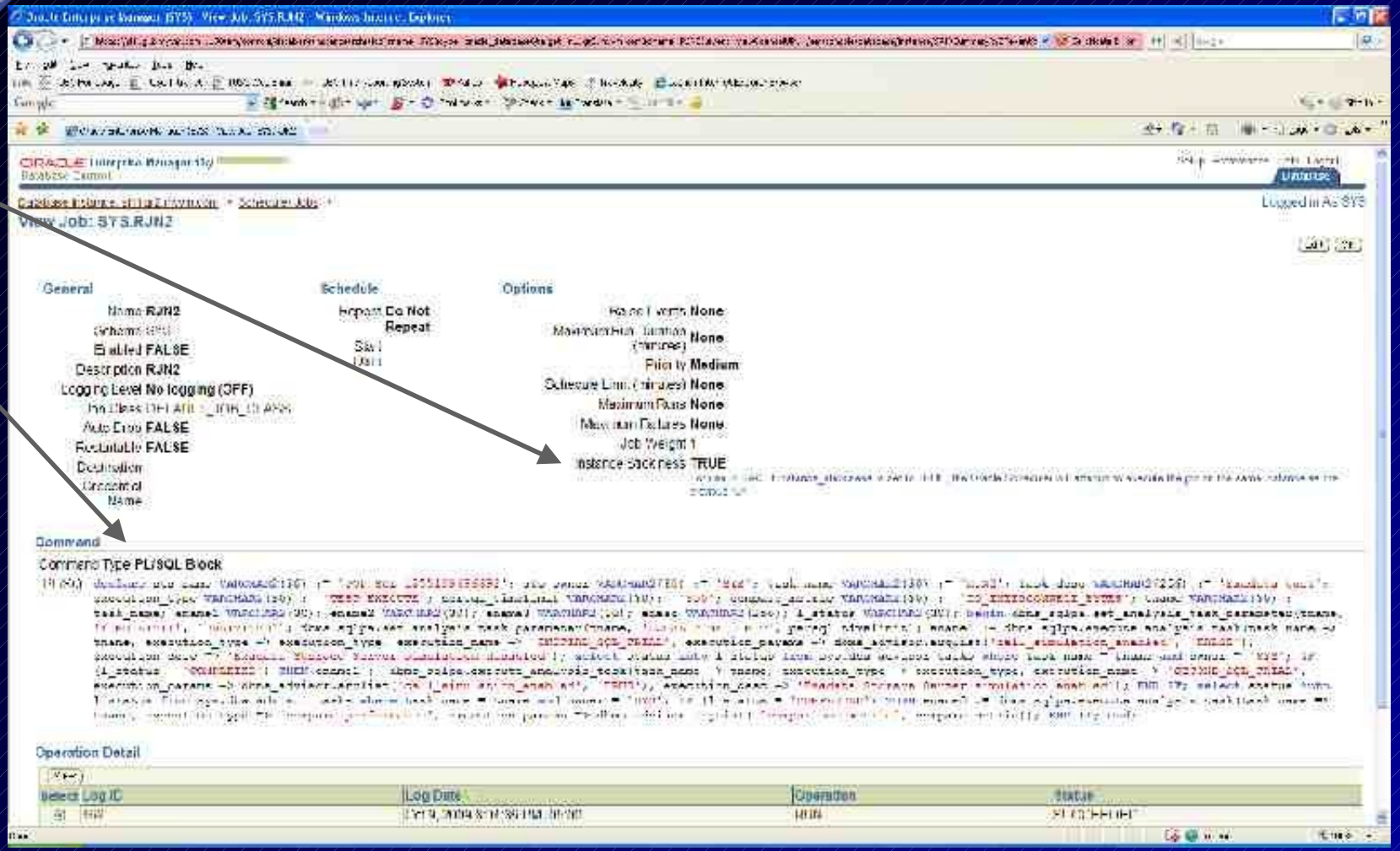
Job is running



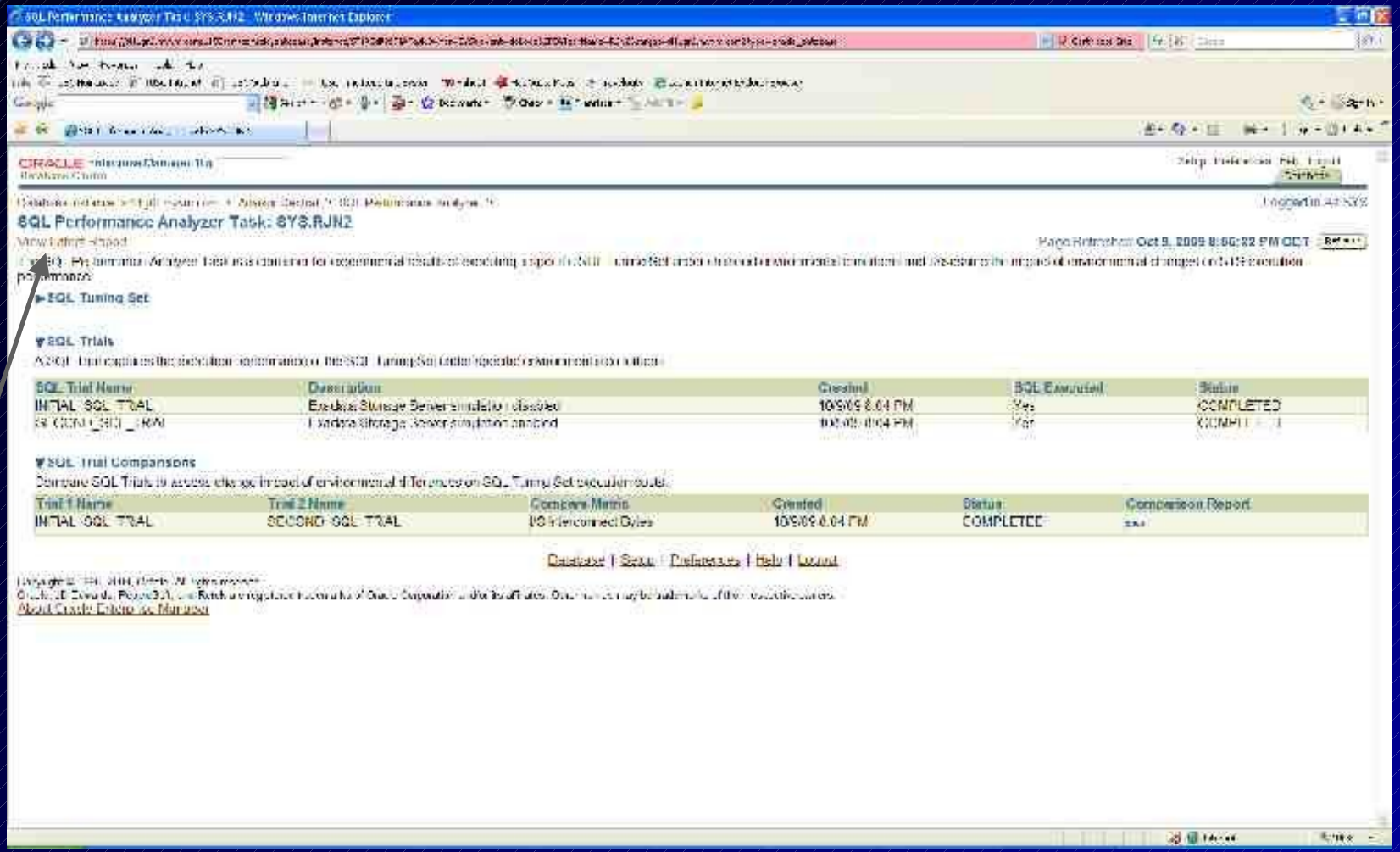
SQL Performance Analyzer 11gR2 – Exadata Simulation

New!

View PL/SQL



SQL Performance Analyzer 11gR2 – Exadata Simulation

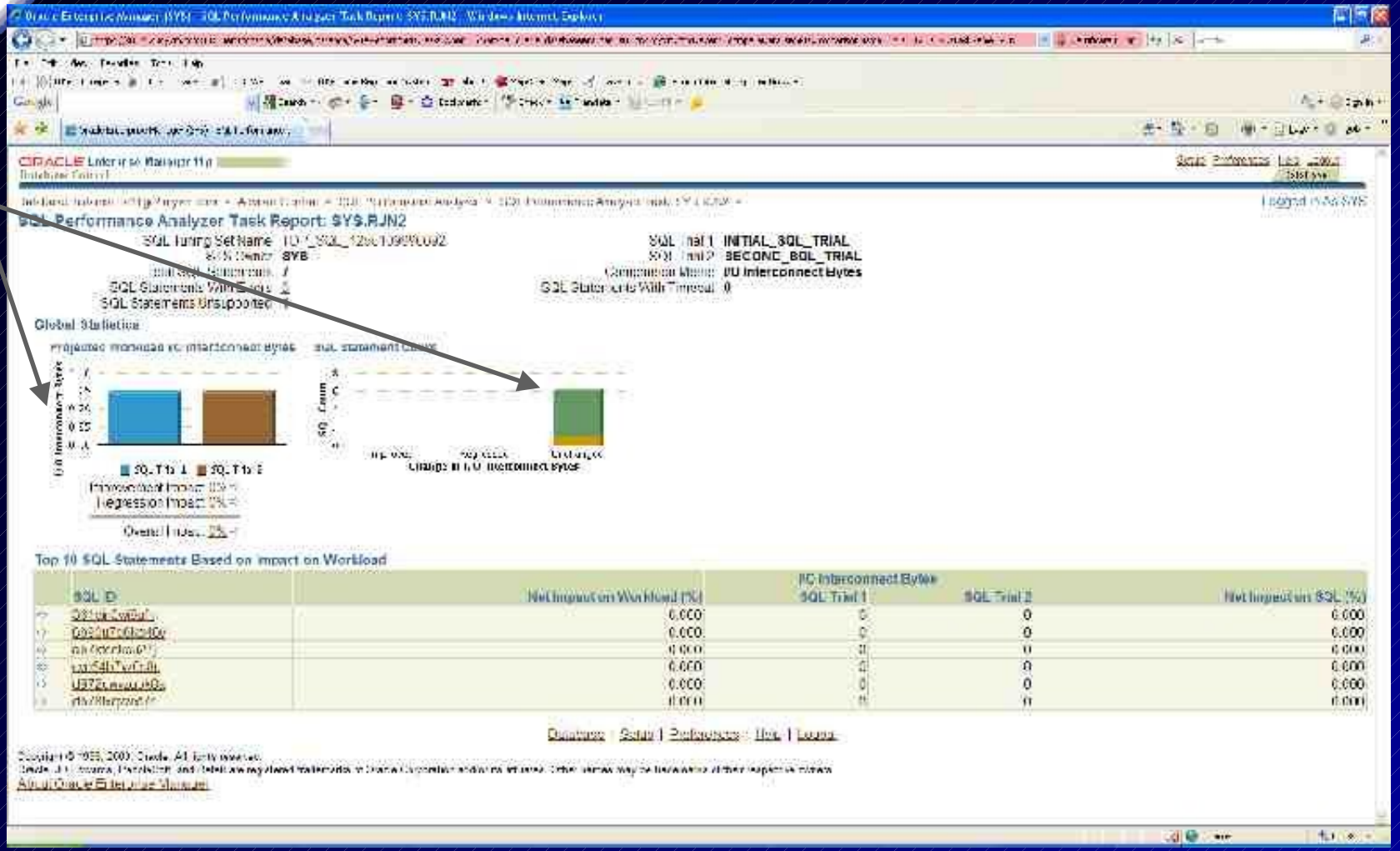


Click on Job after complete

View Report

SQL Performance Analyzer 11gR2 – Exadata Simulation

Simple Job so no benefit

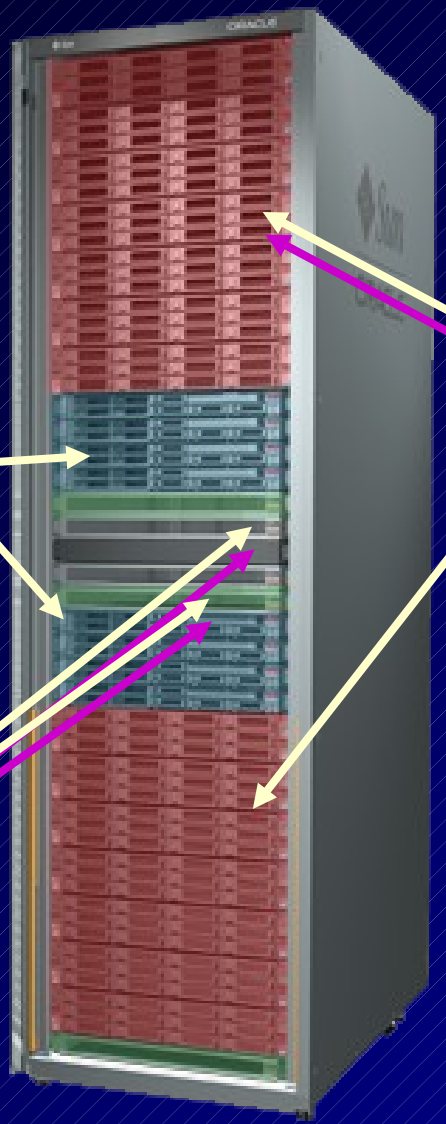


Exadata = Paradigm Shift!

Easier way – Oracle's picture of the Sun Oracle Database Machine

8 Compute Servers

- 8 x 2 sockets x 4 cores = 64 cores
- 576 GB DRAM



14 Storage Servers

- 14x12=168 Disks
- 100T SAS or
- 336T SATA

- 5TB+ flash storage!

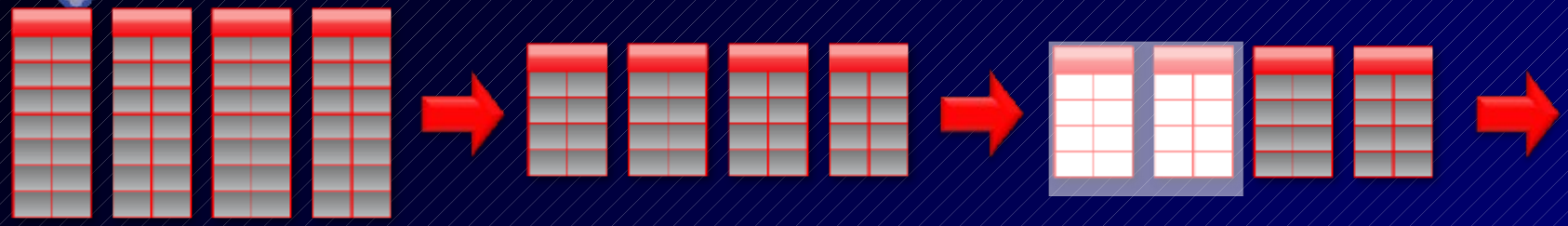
InfiniBand Network

- 40 Gb/sec each direction
- Fault Tolerant

How they got these NUMBERS?

- 8 compute servers (x4170's)
  - 8 servers x 2 CPU sockets x 4 cores = 64 cores (E5540 2.53 GHz)
- 8 servers x 72G DRAM = 576G DRAM (400G useable)
- 14 Storage Servers total 336G DRAM = 912G Total DRAM
- 3 InfiniBand Switches x 36 ports = 108 ports
- 14 Storage Servers (100-336T) with Flash Cache (5T+)
  - 96G x 4 banks = 394G flash cache per storage server
  - 14 storage servers x 394G = 5.376T Flash Cache
  - 12 disks per storage server x 14 servers = 168 disks
  - 168 disks x 600G SAS = 101T SAS
  - 168 disks x 2T SATA = 336T SATA
  - Additional total storage of 4.672T on Database Servers (146G drives)
- 14 storage servers x 2 quad core E5540 = 112 additional cores

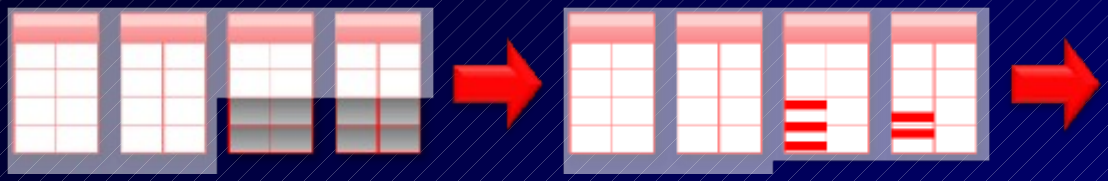
Benefits Multiply*



10 TB of user data
Requires 10 TB of IO

1 TB
with compression

100 GB
with partition pruning



20 GB
with Storage Indexes

5 GB
with Smart Scans

Sub second
On Database Machine

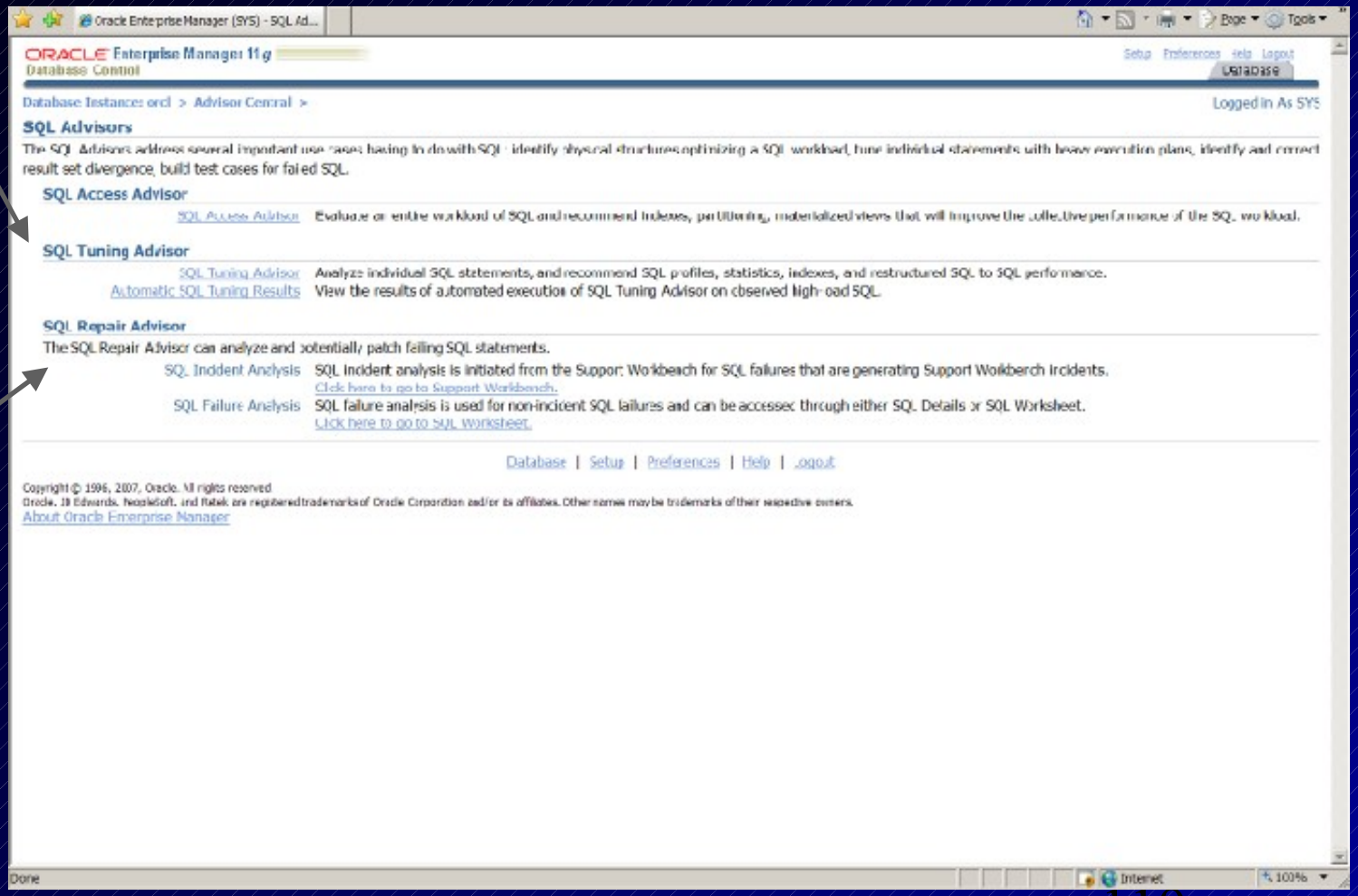
Data is 10x Smaller, Scans are 2000x faster

*Oracle Slide –

SQL Advisors

Tuning Advisors

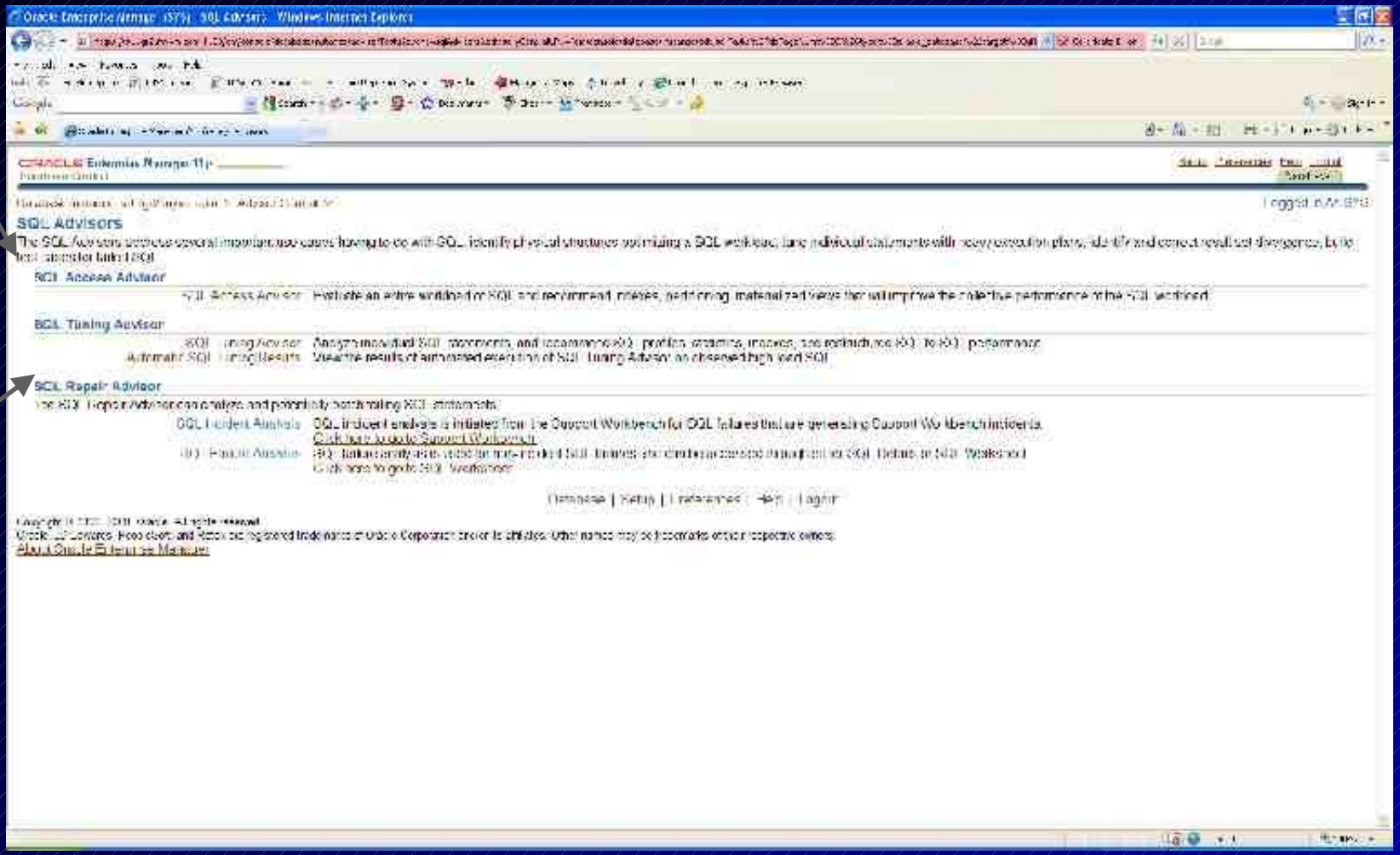
Repair Advisor (next)



SQL Advisors
11gR2 – (same)

Tuning Advisors

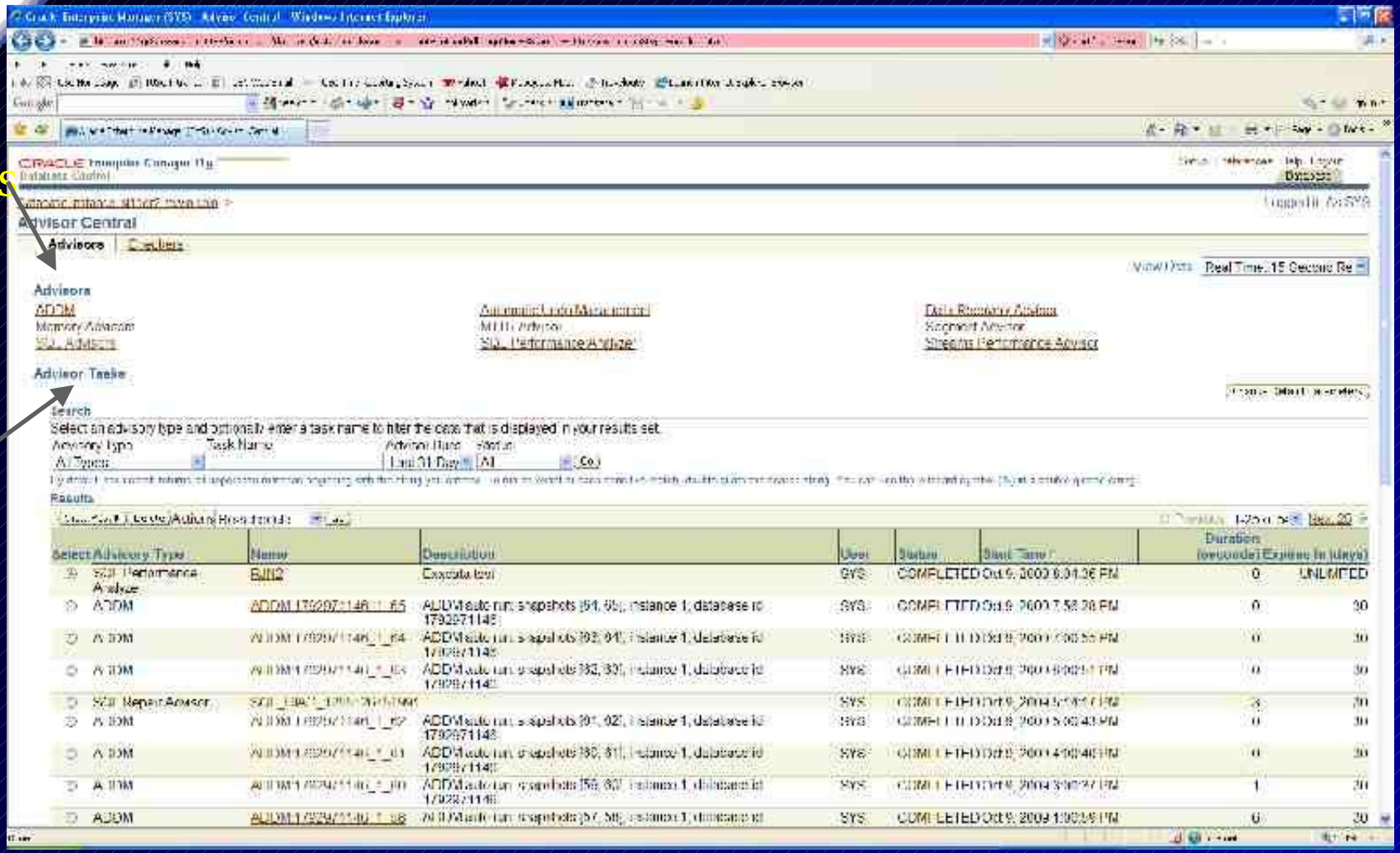
Repair Advisor (next)



Advisor Central
11gR2 – (different)

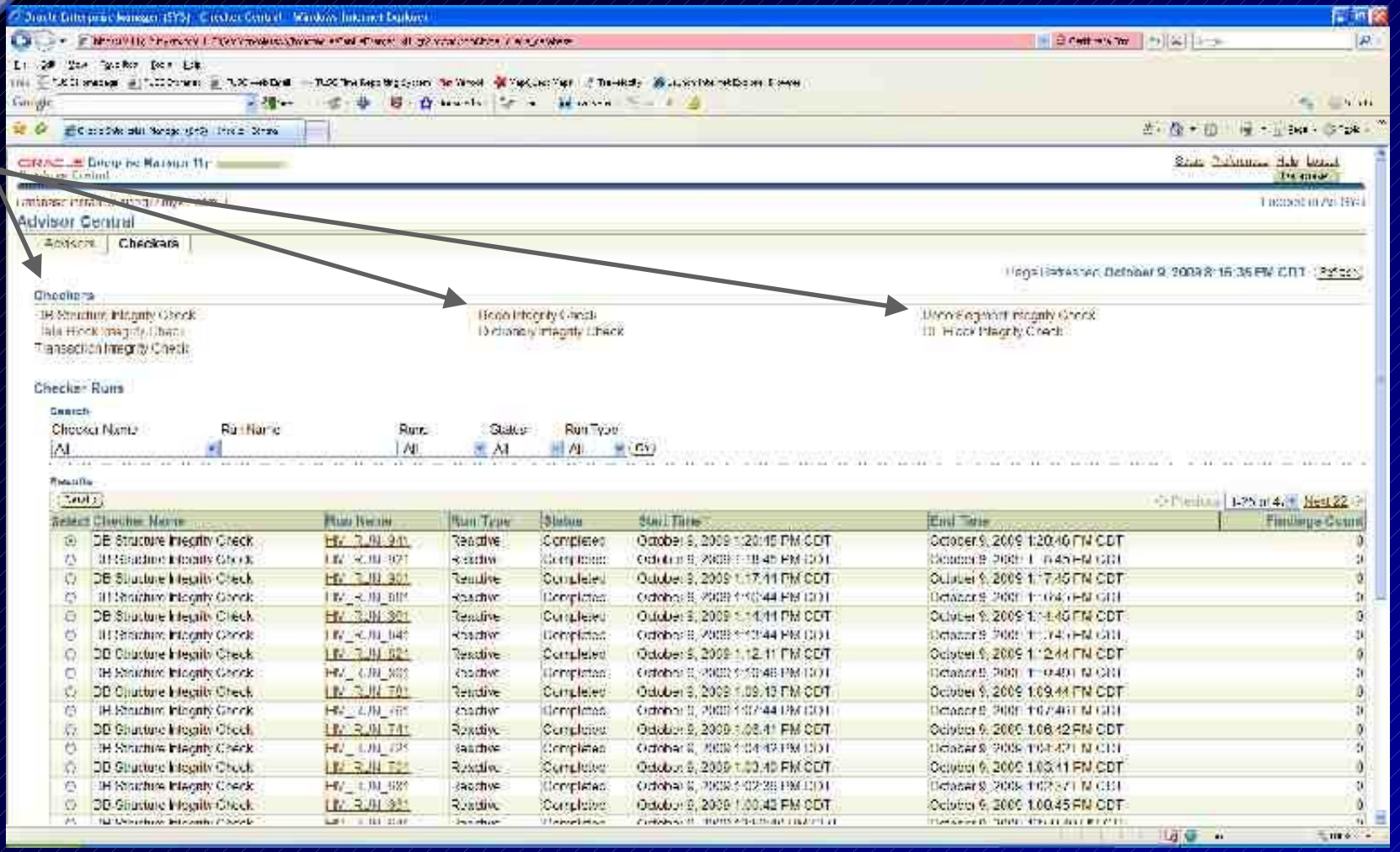
Tuning Advisors

Repair Advisor (next)



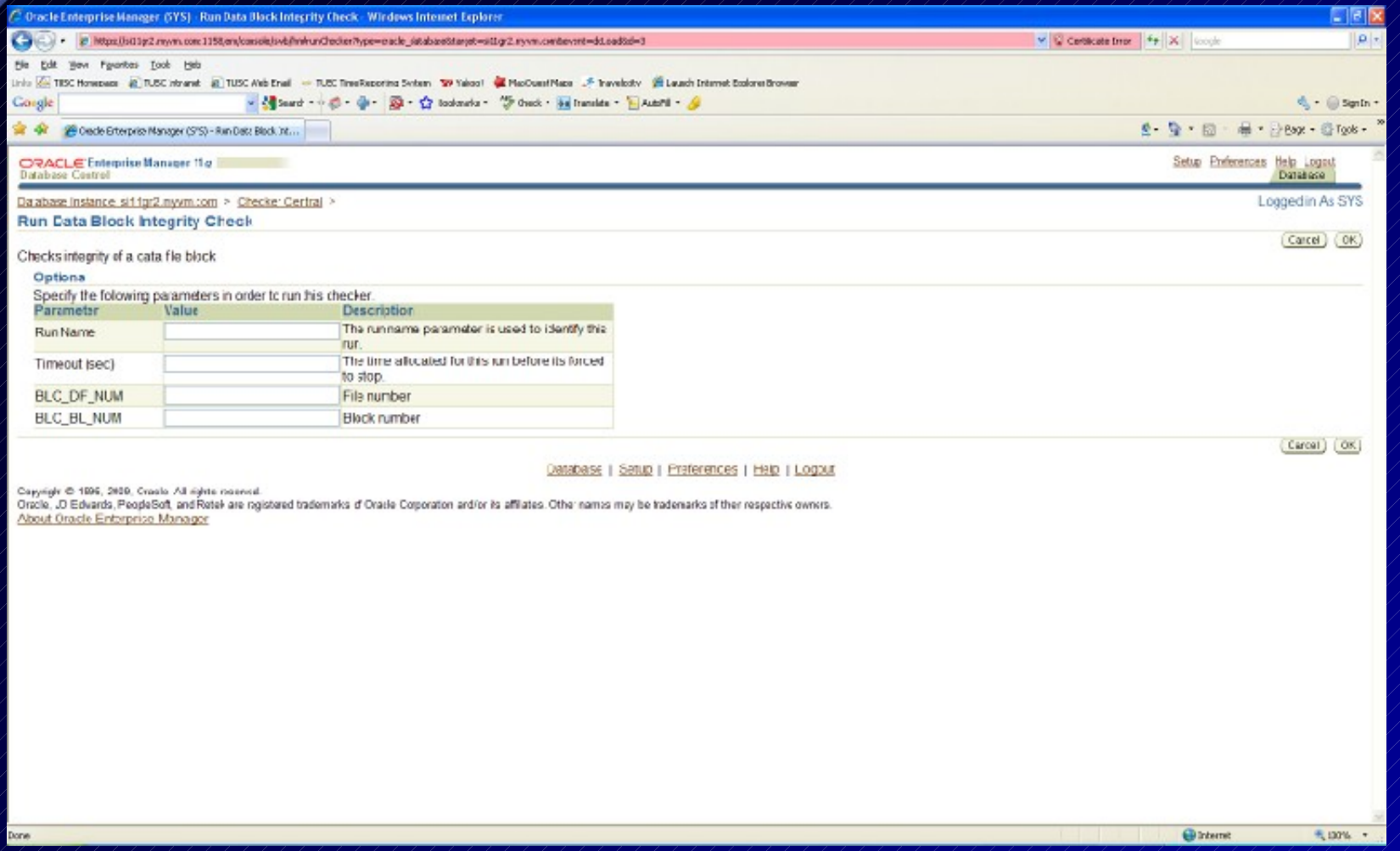
Advisor Central - Checkers
11gR2 – (different)

Integrity Checks



Advisor Central - Checkers
11gR2 – (different)

Data Block: Integrity Check



SQL Access Advisor
NEW Partition Advisor

SQL Access Advisor & NEW Partition Advisor

- The SQL Advisor now combines the functionality of the SQL Tuning Advisor, SQL Access Advisor and the new Partition Advisor.
  - Recommends Partitioning Needs
  - Utilize a previous SQL Tuning Set
  - Take SQL straight from what's currently in the CACHE.
  - Create a hypothetical workload
  - SQL Access Advisor checks Indexes, Partitions or Materialized Views (**schema related issues**)

SQL Access Advisor & NEW Partition Advisor

Step One

Use a SQL Tuning Set

SQL Access Advisor & NEW Partition Advisor

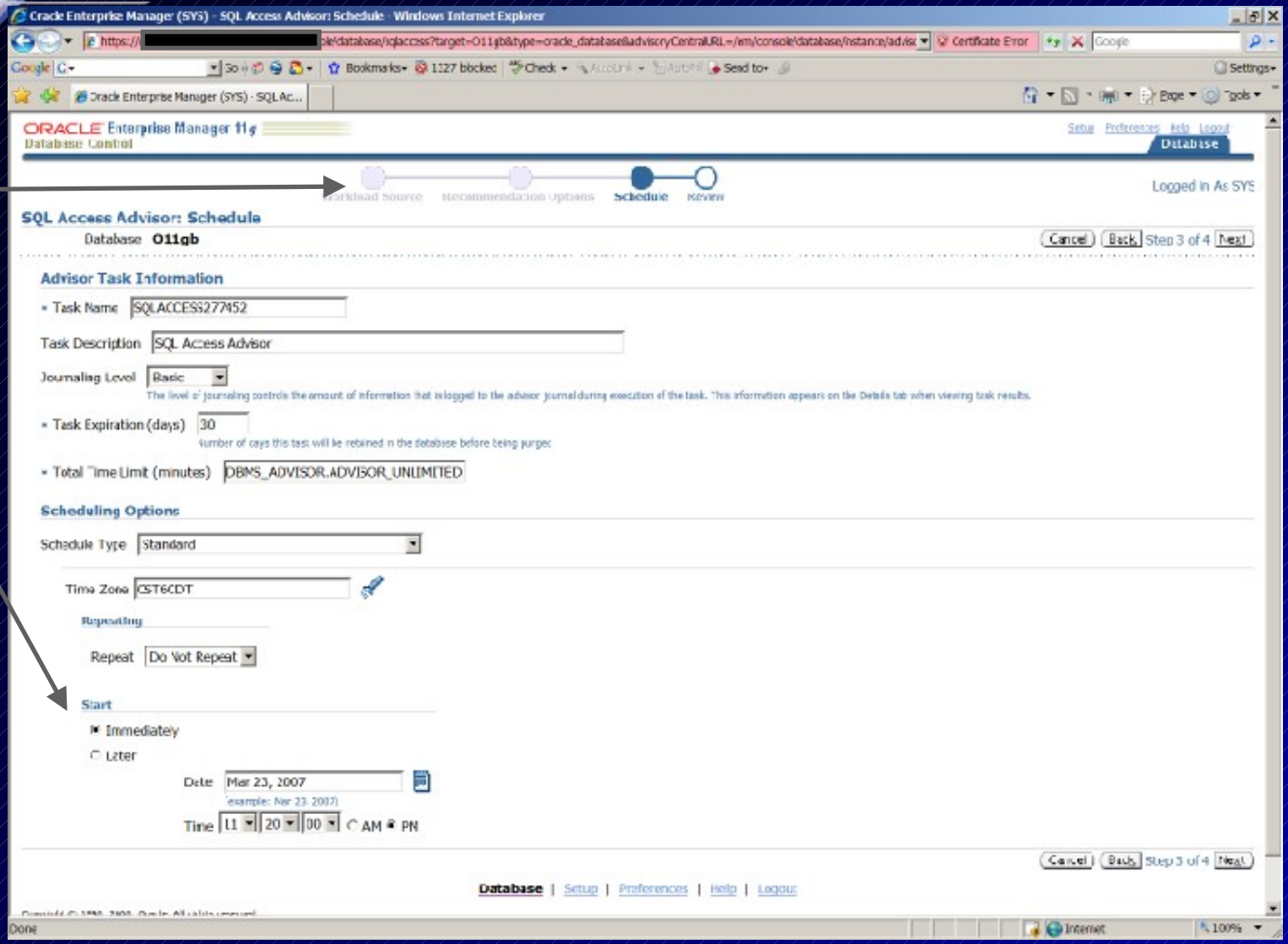
Look at Partitions

Quick Solution

SQL Access Advisor & NEW Partition Advisor

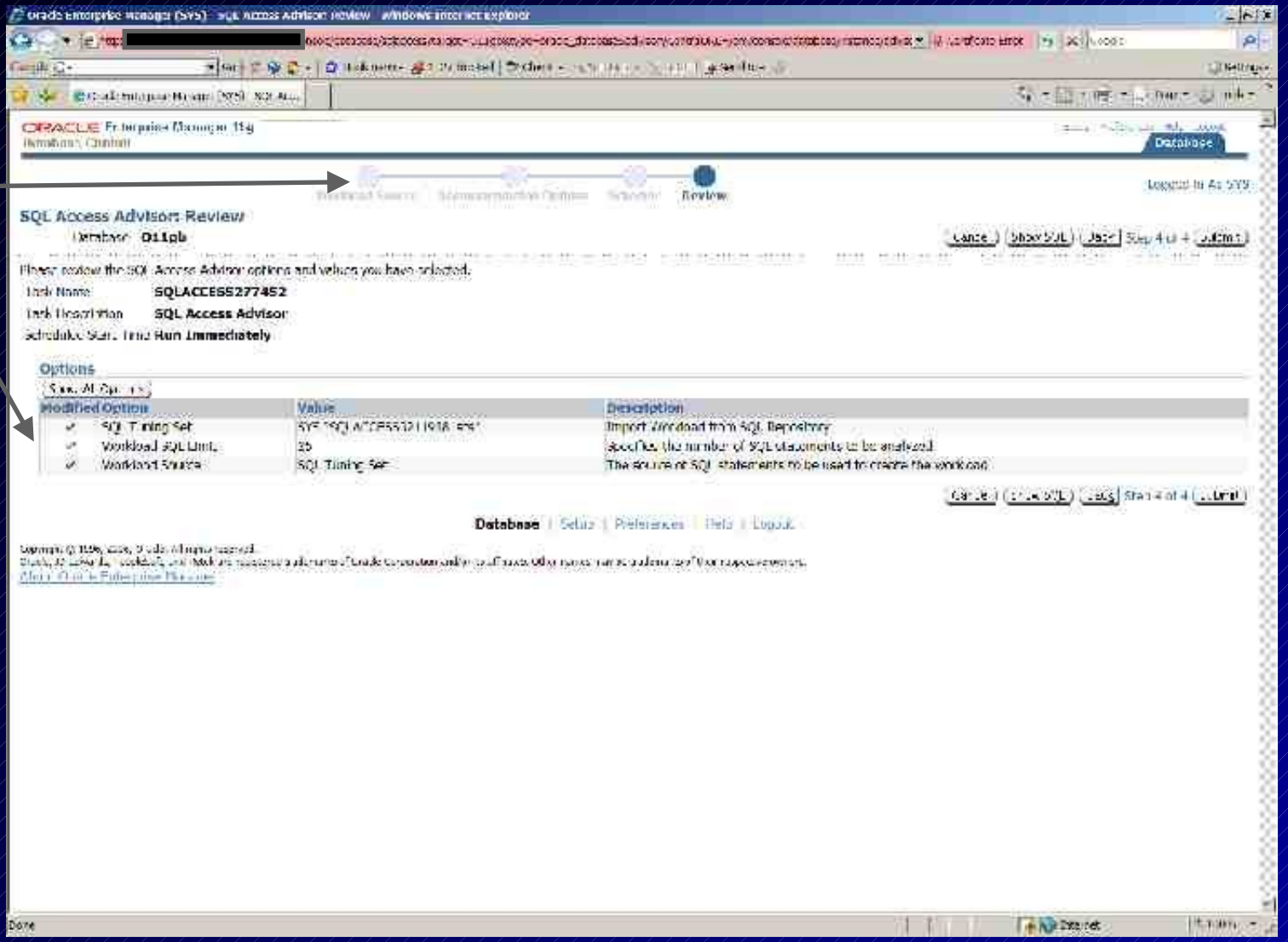
Step 3 Schedule it.

Run it Now!



SQL Access Advisor & NEW Partition Advisor

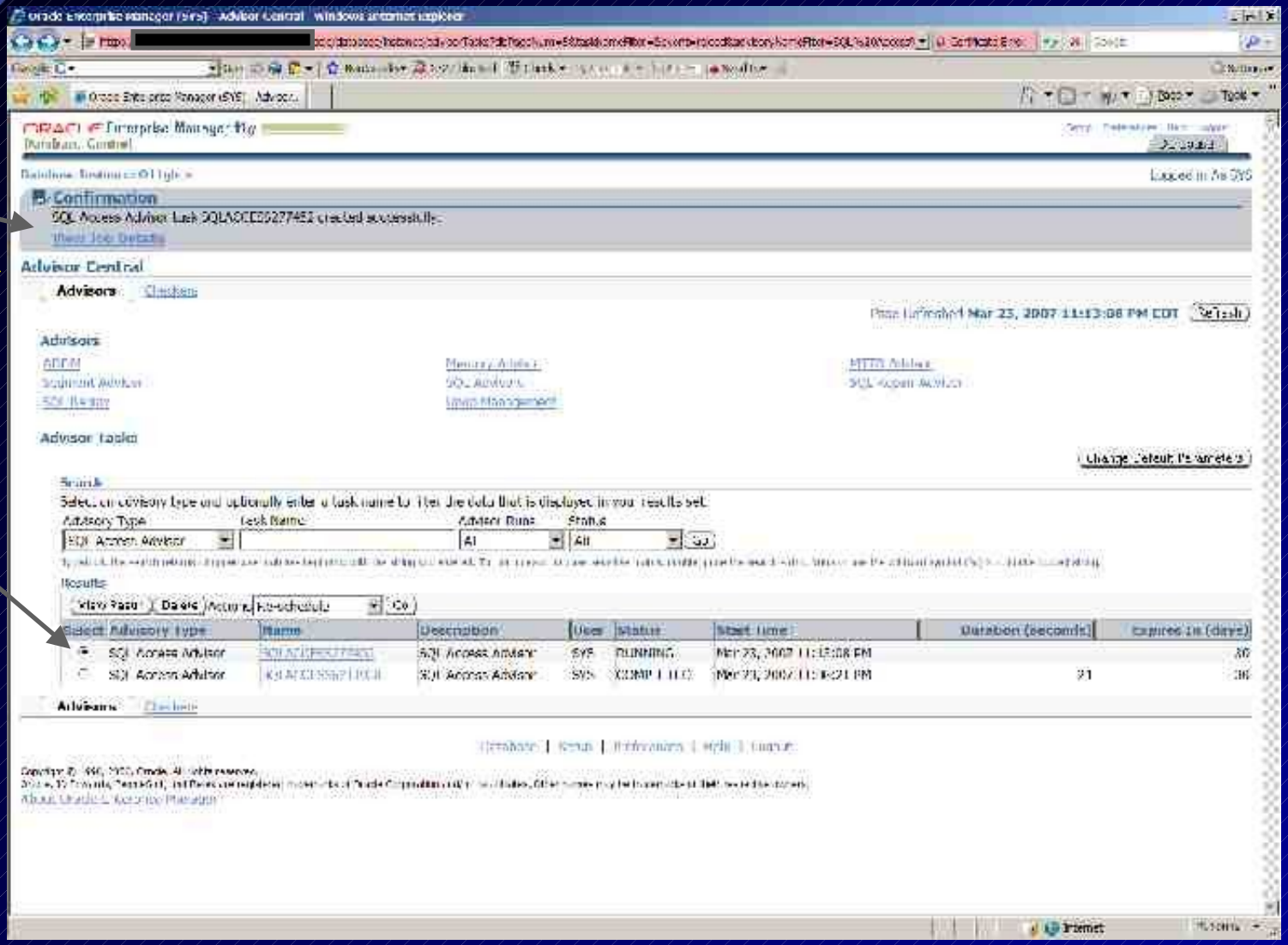
Final Review



SQL Access Advisor & NEW Partition Advisor

Job Submitted

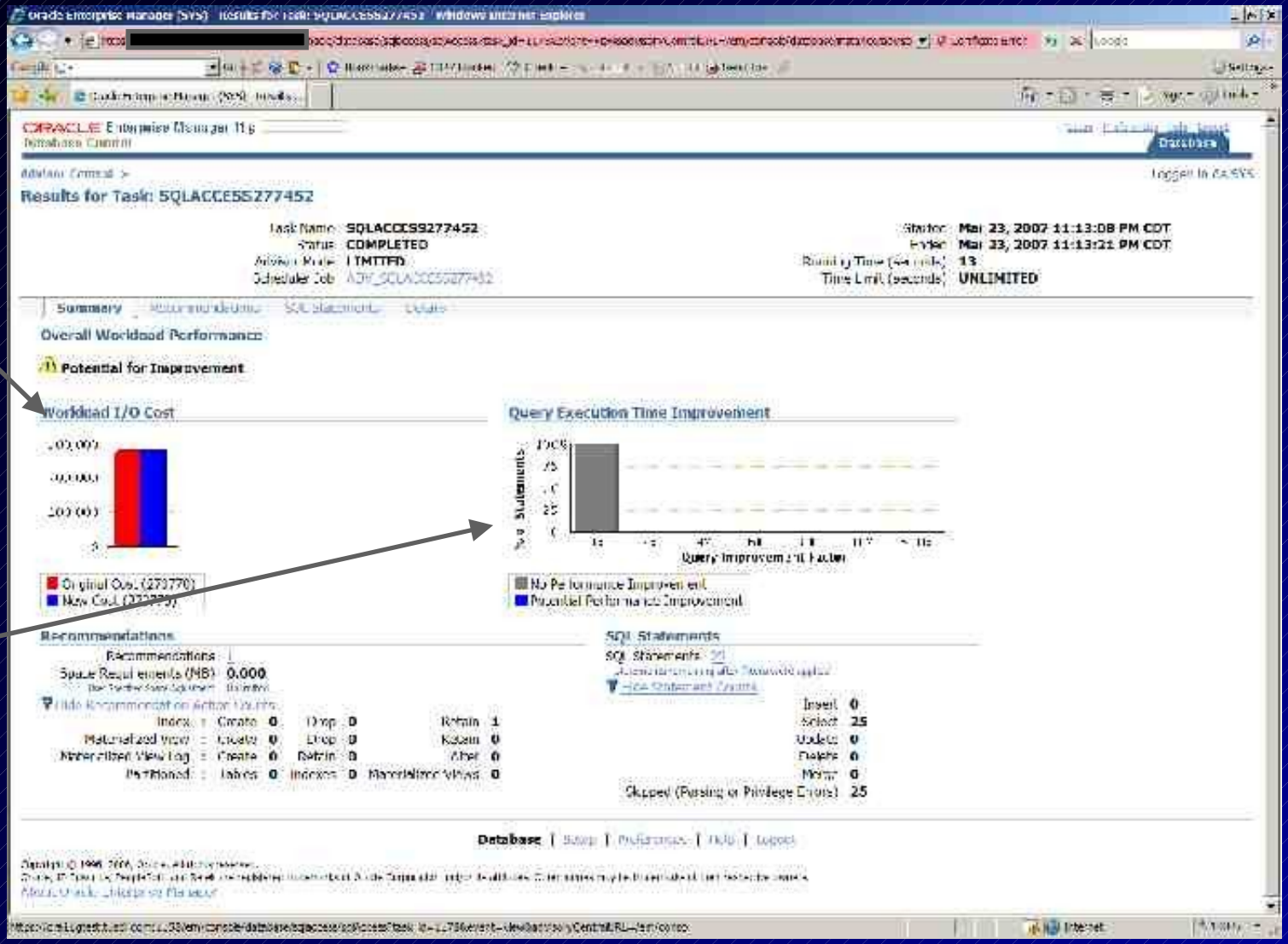
Job Running Now.



SQL Access Advisor & NEW Partition Advisor

Improve I/O

Improve Execution Time



The SQL Repair Advisor

ORA-600

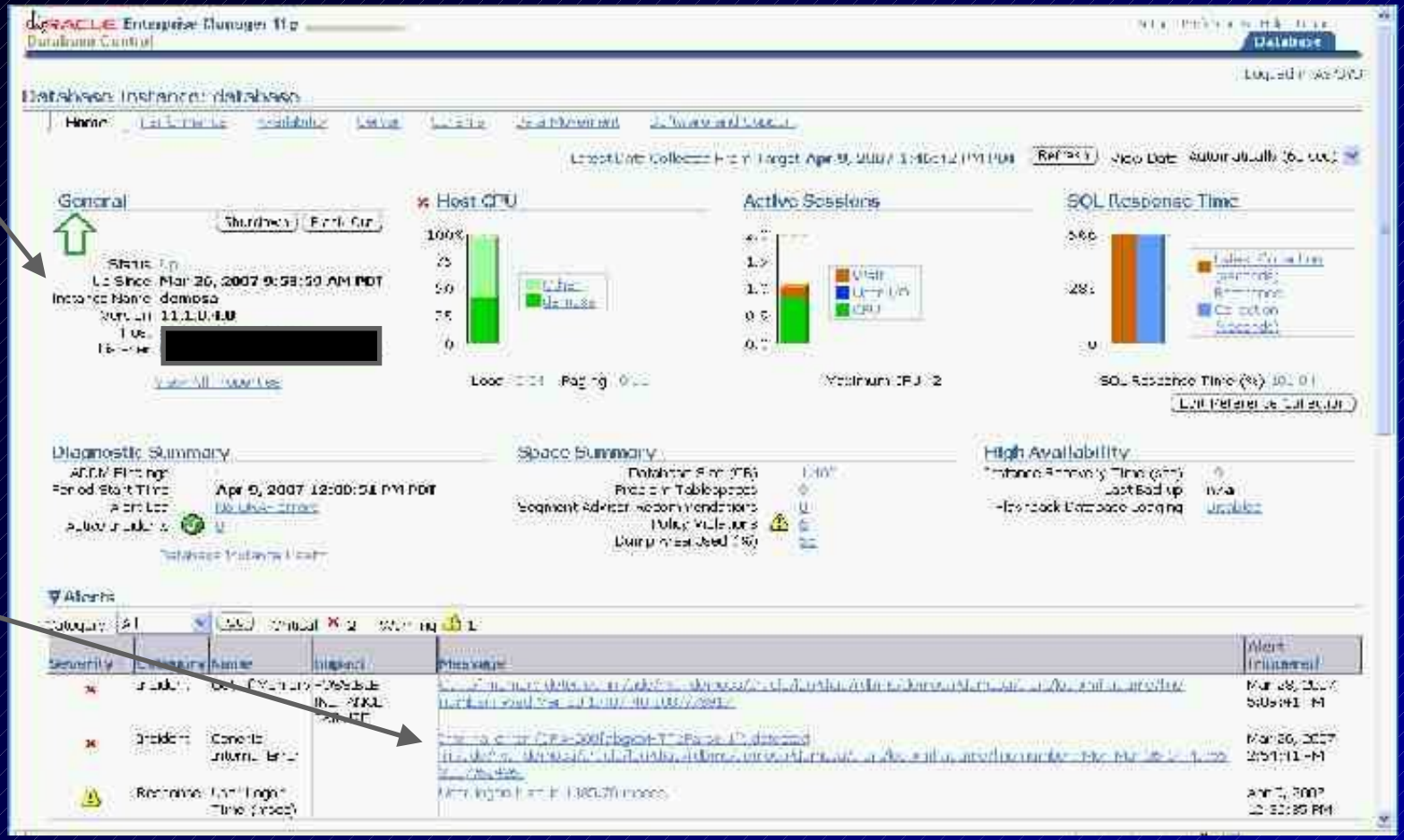
SQL Repair Advisor

- Used to Repair Problem SQL – Oracle Errors
- Reloads and recompiles SQL statements to gather diagnostics information to fix.
- Uses the diagnostic information to repair the problem SQL statement (DBMS_SQLDIAG)
- Will fix error **going through compilation, execution and trying different routes** (could be a slower route for now) to come up with a temporary SQL Patch without error until fixed.

SQL Repair Advisor – Go straight from Alerts

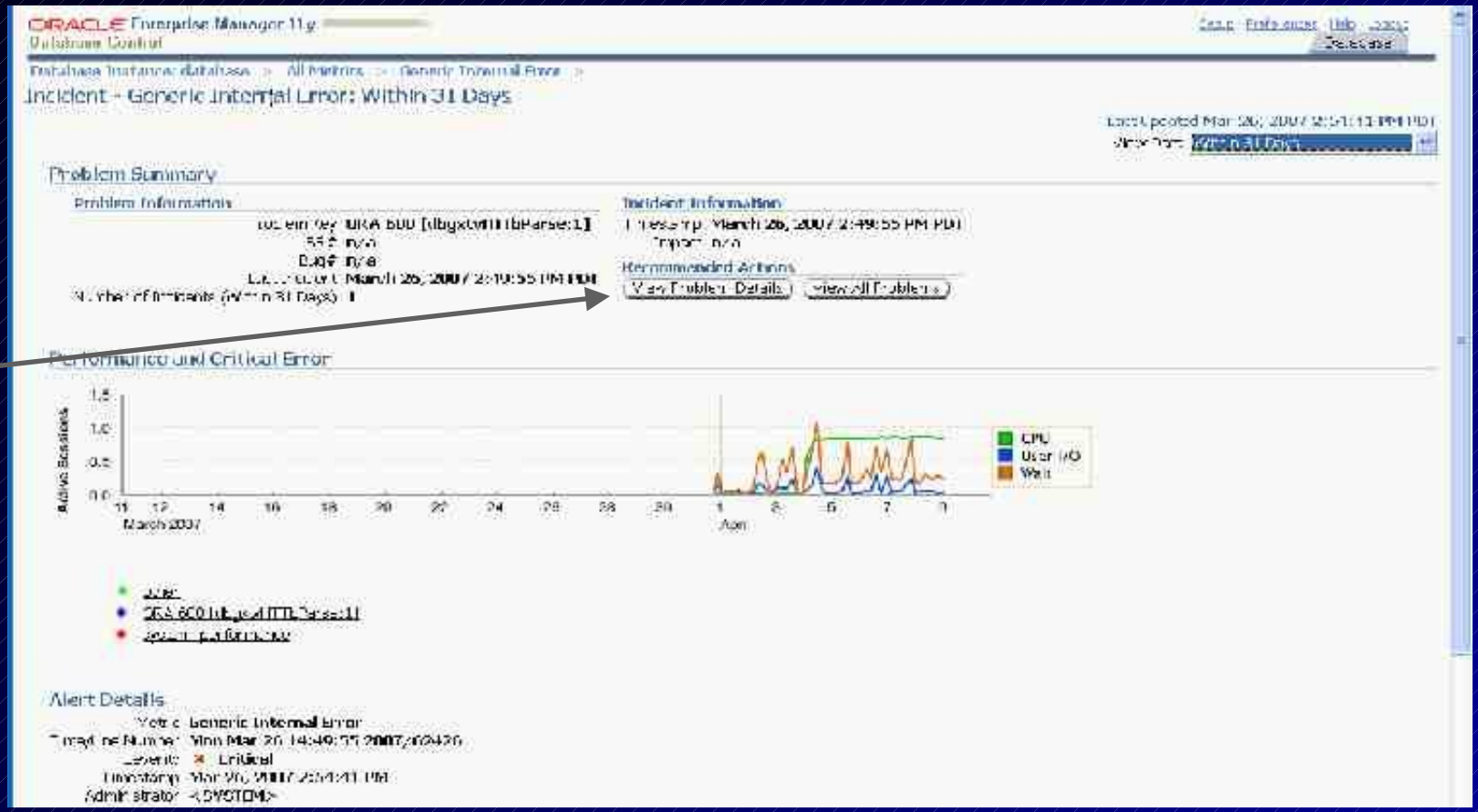
Go to the Database Instance

Click Alert (ORA-600) message text to see details



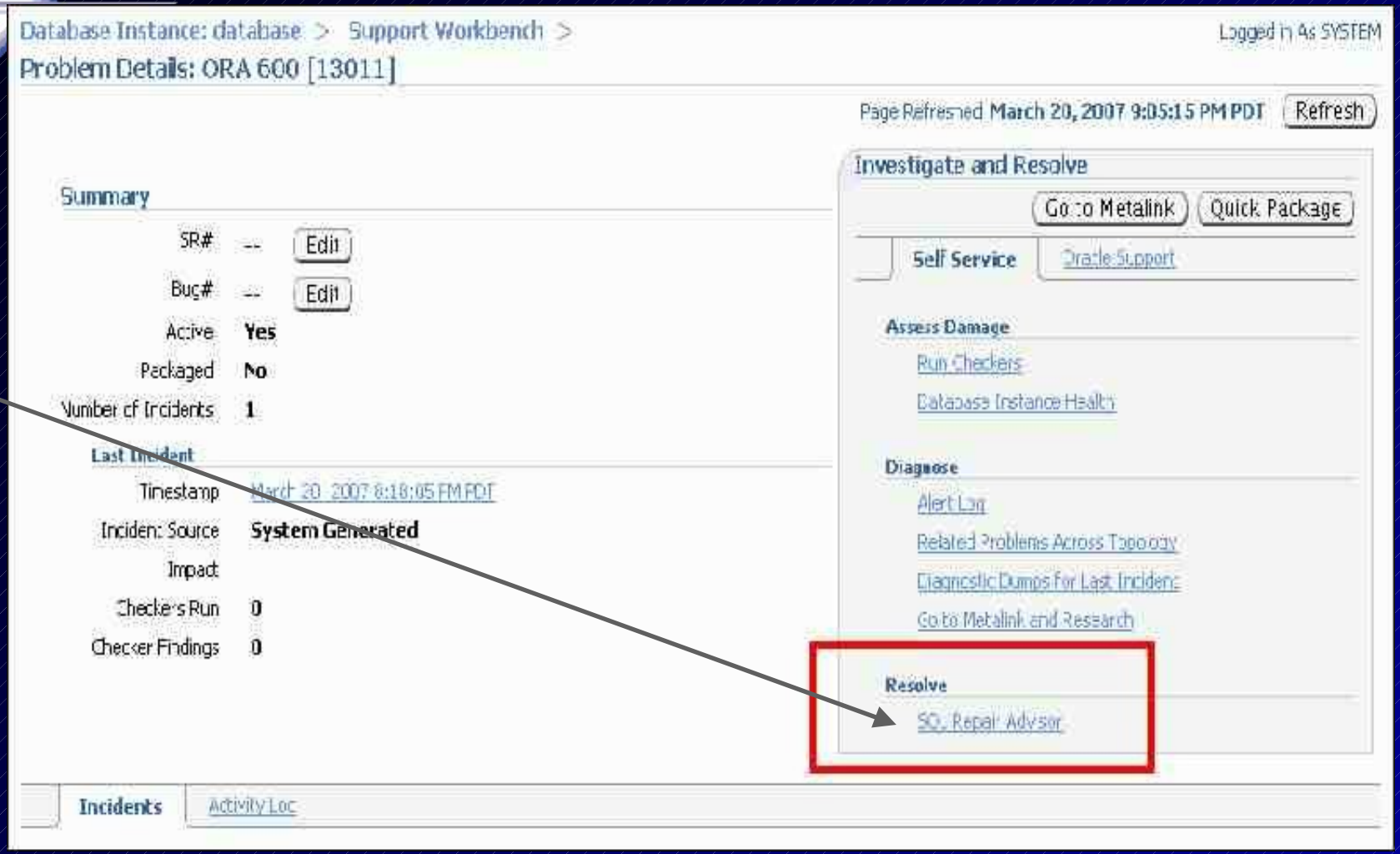
SQL Repair Advisor – View Problem Details

Click on View Problem Details to go to the Support Bench



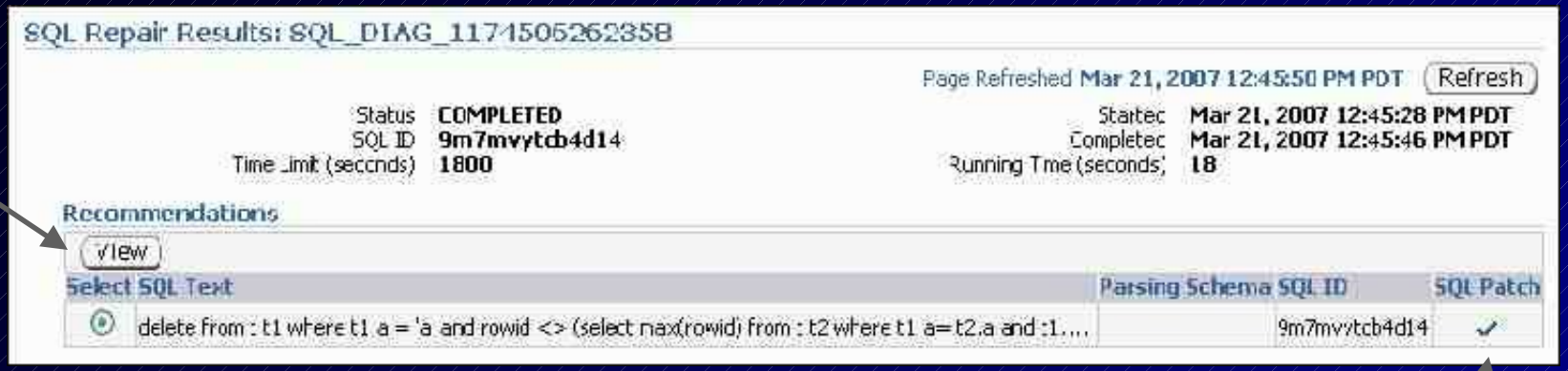
Support Workbench - Details

Click on SQL Repair Advisor



Results from SQL Repair Advisor

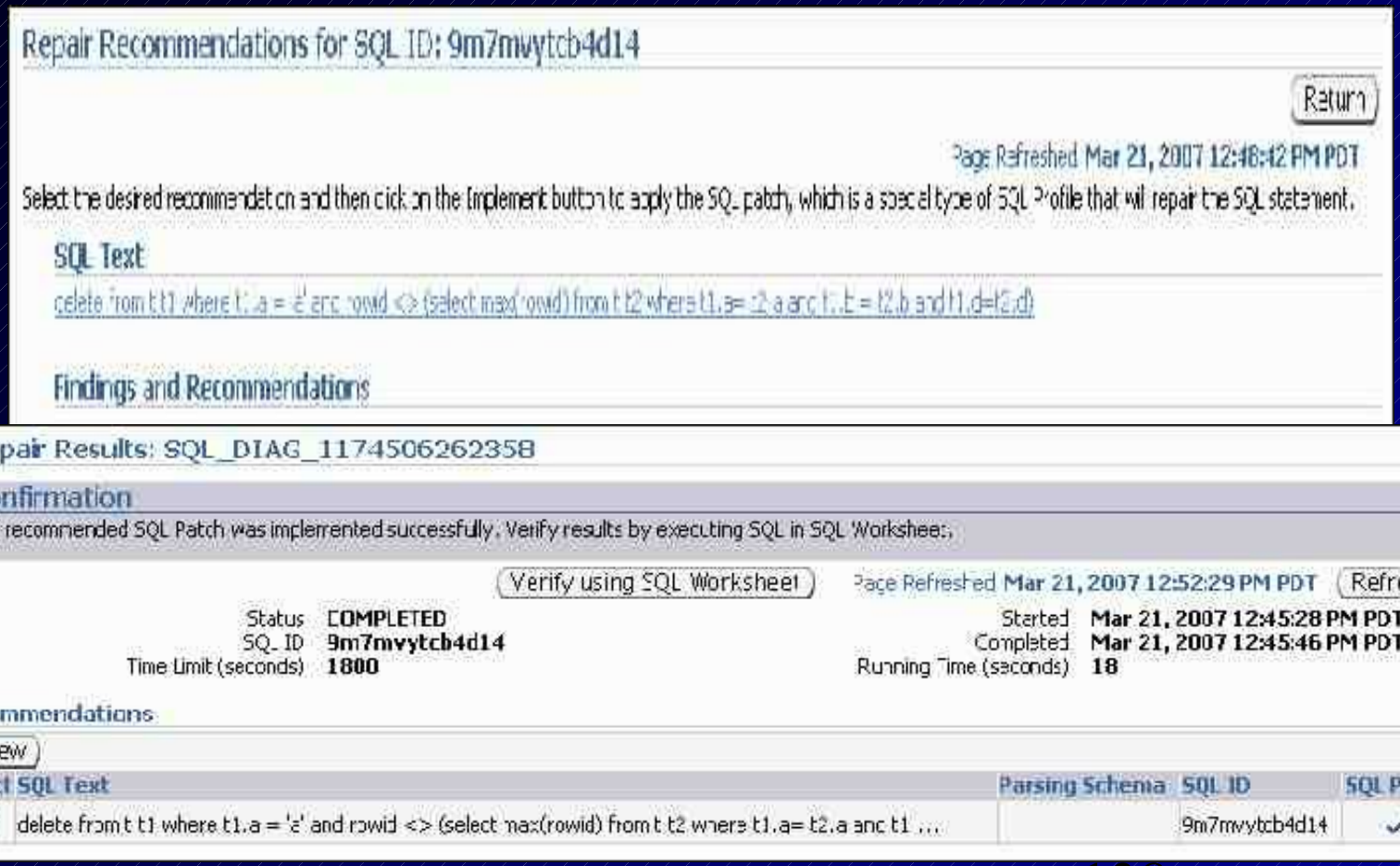
Click on View to Get the Detail finding of the Advisor



Note a SQL Patch (FIX for the SQL) has been generated

SQL Repair Advisor
Recommendation / Confirmation

Click on Implement To accept the SQL Patch



Real Application Testing!
Database workload capture and replay

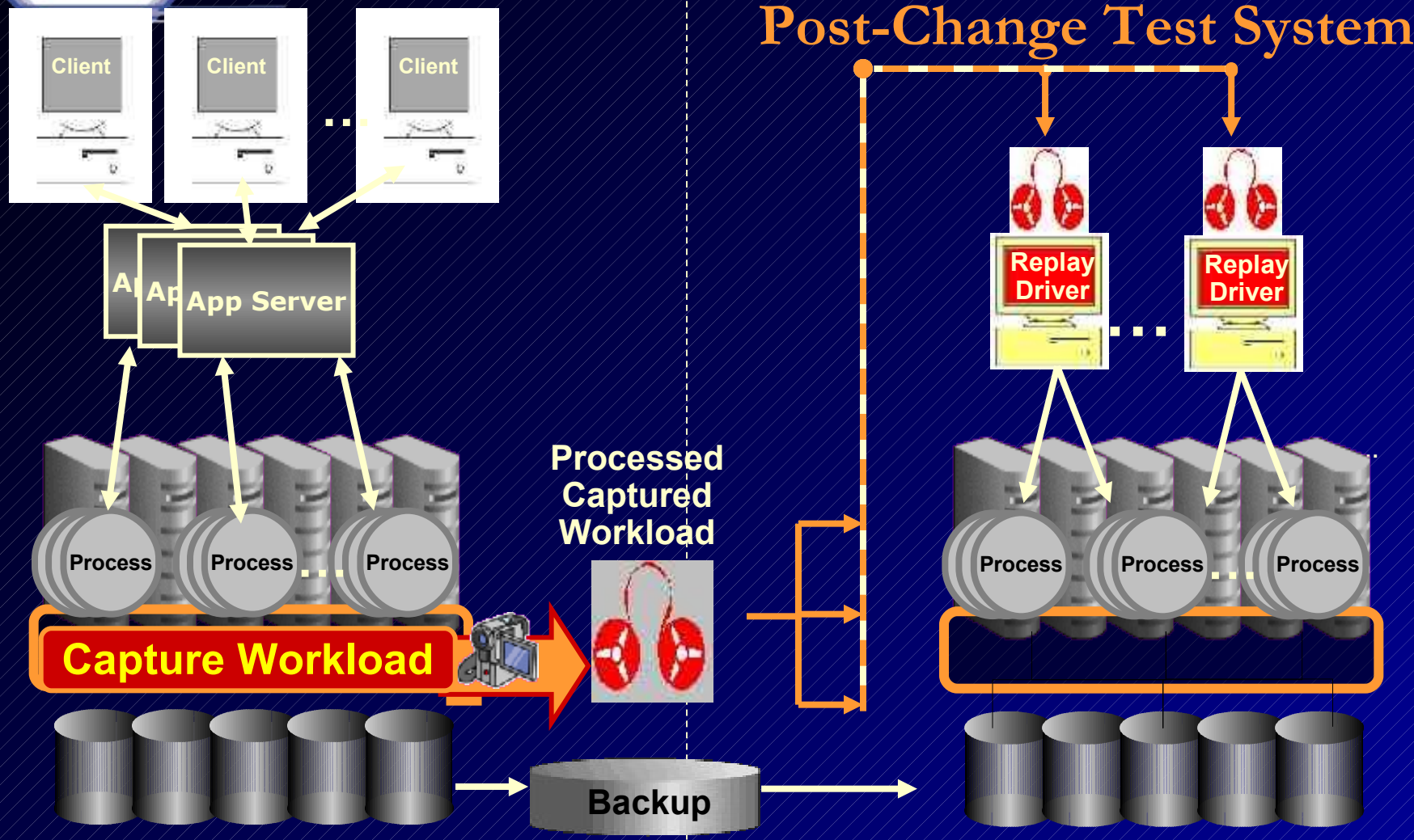
Database workload capture and replay

- Used to capture **database workload** on one system and replay later on a different system. Useful to compare two different systems.
- Could rival LoadRunner in the future (may be more precise!)

Brief Steps:

- Capture workload on a database even from 10gR2
- Restore the database on a test system to the SCN when capture begins
- Perform upgrade and make changes to the test system as needed
- Preprocess the captured workload if it is not preprocessed
- Configure the test system for replay (I don't do this here)
- Replay workload on the restored database (I don't have this in this presentation, but will show some of the screens to do it)
- Great to test upgrade to 11g **(Capture 10gR2 then test against 11g)**

Pre-Change (could be 9.2.0.8 or 10g Capture) Production System

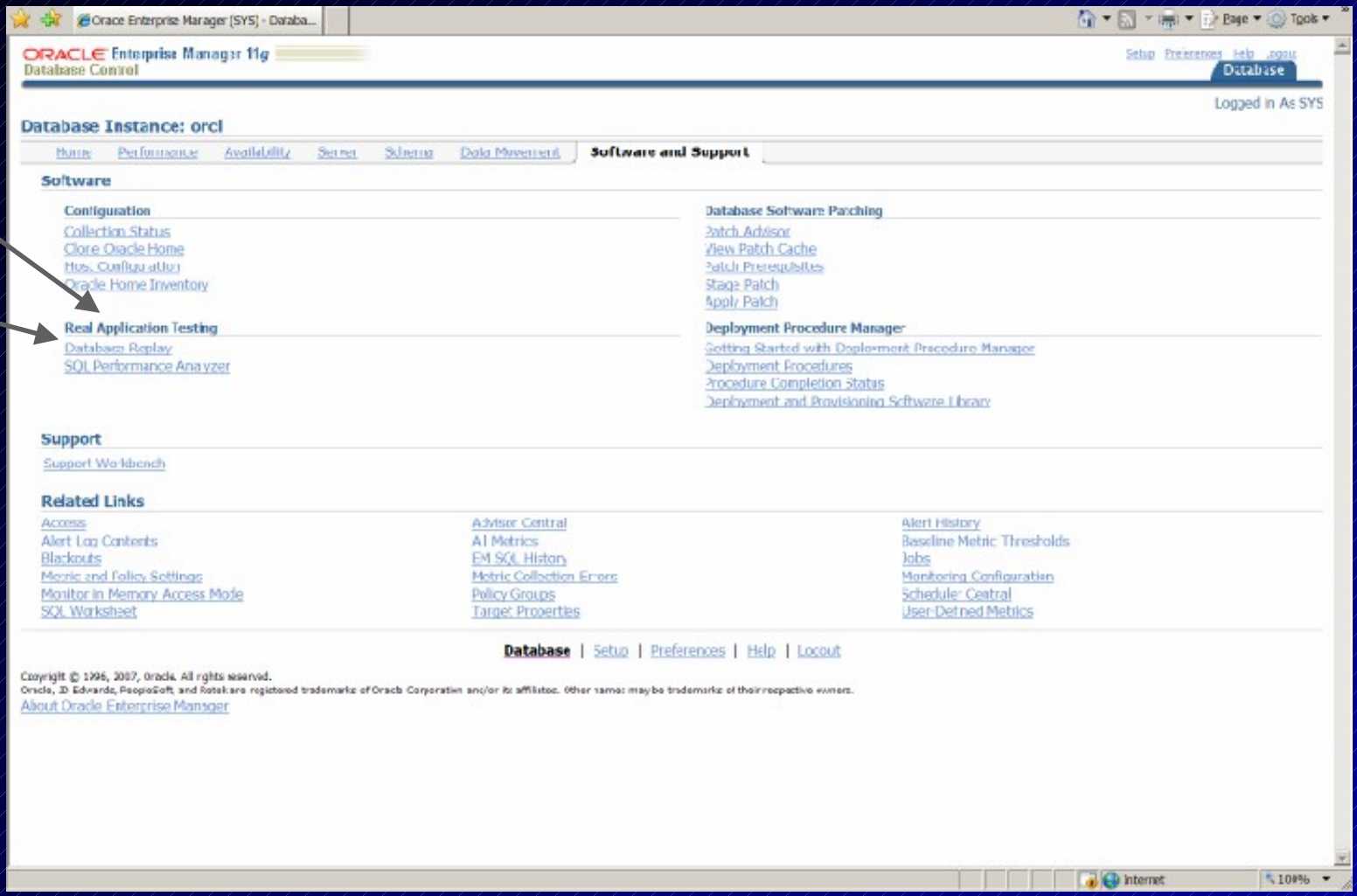


Can use Snapshot Standby as test system

Database Replay
FYI Only – Download to view in detail

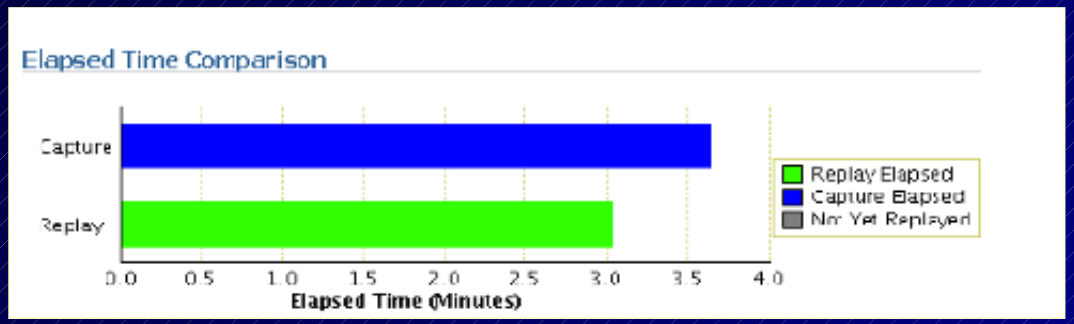
Real App Testing:

Database Replay



Replay Options...

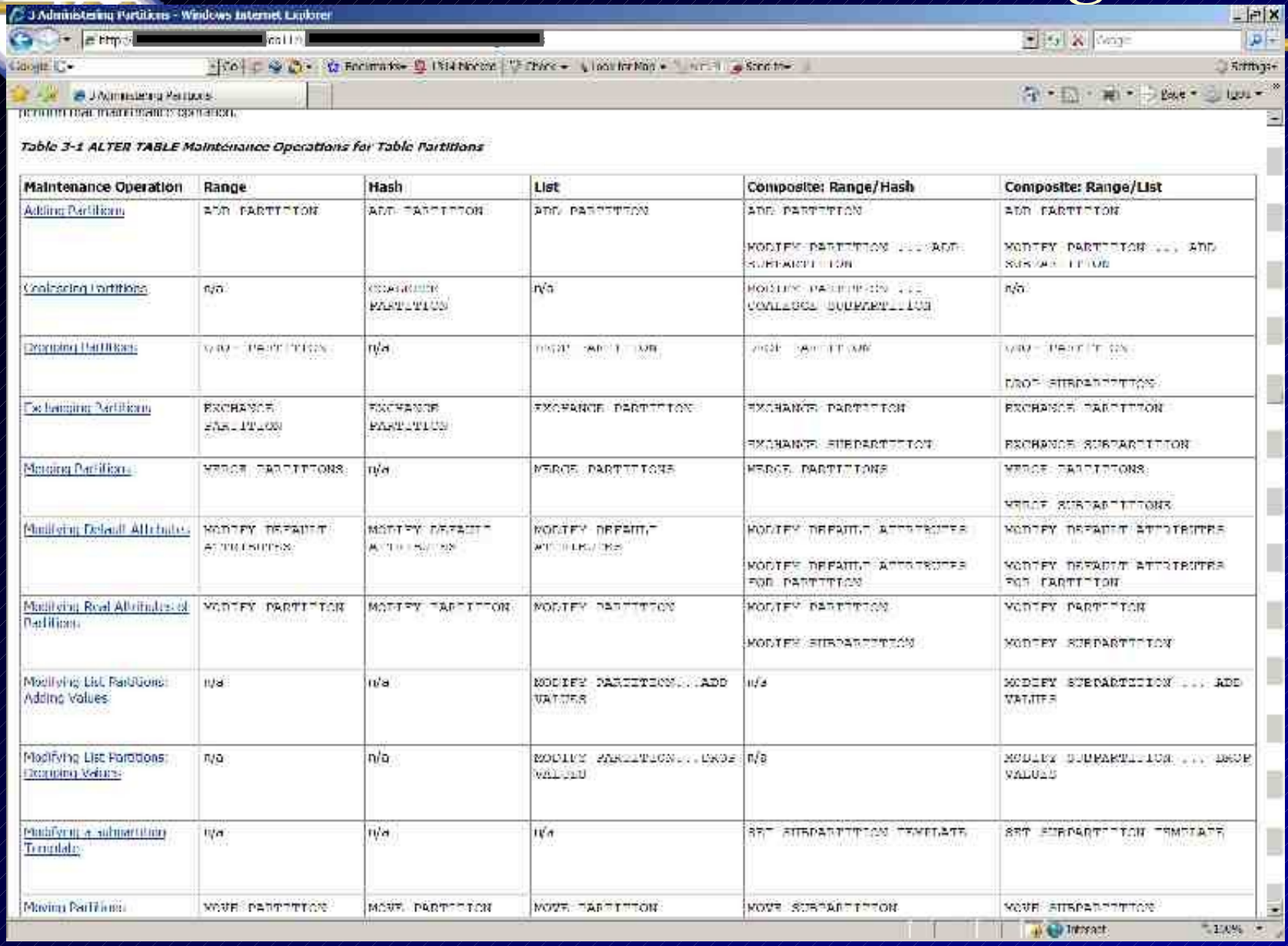
- Synchronized Replay
  - Exact Concurrency, commits & data divergence minimal
- Unsynchronized Replay
  - Not the same concurrency or commits
  - Data divergence can be large depending on load test performed
- Creates Report
  - Data Divergence
  - Error Divergence
  - Performance Divergence



Partitioning: (FYI Only)

- Tables can be split into many pieces (10g).
- Only a subset of the data is queried
- All of the data COULD be queried
- Leads to enhanced performance of large tables
- Re-orgs & backups can be done on a partition level
- 4 quick examples follow (many many rules for each)
- WHAT'S NEW IN ORACLE 11G

The Rules – See Partitioning Guide



Administering Partitions - Windows Internet Explorer

Table 3-1 ALTER TABLE Maintenance Operations for Table Partitions

Maintenance Operation	Range	Hash	List	Composite: Range/Hash	Composite: Range/List
Adding Partitions	ADD PARTITION	ADD PARTITION	ADD PARTITION	ADD PARTITION MODIFY PARTITION ... ADD SUBPARTITION	ADD PARTITION MODIFY PARTITION ... ADD SUBPARTITION
Coalescing Partitions	n/a	COALESCE PARTITION	n/a	MODIFY PARTITION ... COALESCE SUBPARTITION	n/a
Dropping Partitions	DROP PARTITION	n/a	DROP PARTITION	DROP PARTITION	DROP PARTITION DROP SUBPARTITION
Exchanging Partitions	EXCHANGE PARTITION	EXCHANGE PARTITION	EXCHANGE PARTITION	EXCHANGE PARTITION EXCHANGE SUBPARTITION	EXCHANGE PARTITION EXCHANGE SUBPARTITION
Merging Partitions	MERGE PARTITIONS	n/a	MERGE PARTITIONS	MERGE PARTITIONS	MERGE PARTITIONS MERGE SUBPARTITIONS
Modifying Default Attributes	MODIFY DEFAULT ATTRIBUTES	MODIFY DEFAULT ATTRIBUTES	MODIFY DEFAULT ATTRIBUTES	MODIFY DEFAULT ATTRIBUTES MODIFY DEFAULT ATTRIBUTES FOR PARTITION	MODIFY DEFAULT ATTRIBUTES MODIFY DEFAULT ATTRIBUTES FOR PARTITION
Modifying Real Attributes of Partitions	MODIFY PARTITION	MODIFY PARTITION	MODIFY PARTITION	MODIFY PARTITION MODIFY SUBPARTITION	MODIFY PARTITION MODIFY SUBPARTITION
Modifying List Partitions: Adding Values	n/a	n/a	MODIFY PARTITION...ADD VALUES	n/a	MODIFY SUBPARTITION ... ADD VALUES
Modifying List Partitions: Dropping Values	n/a	n/a	MODIFY PARTITION...DROP VALUES	n/a	MODIFY SUBPARTITION ... DROP VALUES
Modifying a subpartition Template	n/a	n/a	n/a	SET SUBPARTITION TEMPLATE	SET SUBPARTITION TEMPLATE
Moving Partitions	MOVE PARTITION	MOVE PARTITION	MOVE PARTITION	MOVE SUBPARTITION	MOVE SUBPARTITION

Range Partitioning (V8)

```
CREATE TABLE DEPT
(DEPTNO                 NUMBER(2),
DEPT_NAME               VARCHAR2(30))
PARTITION BY RANGE(DEPTNO)
 (PARTITION D1 VALUES   LESS THAN (10) TABLESPACE DEPT1,
PARTITION D2 VALUES     LESS THAN (20) TABLESPACE DEPT2,
PARTITION D3 VALUES     LESS THAN (MAXVALUE) TABLESPACE
DEPT3);
```

```
INSERT INTO DEPT VALUES (1, 'DEPT 1');
INSERT INTO DEPT VALUES (7, 'DEPT 7');
INSERT INTO DEPT VALUES (10, 'DEPT 10');
INSERT INTO DEPT VALUES (15, 'DEPT 15');
INSERT INTO DEPT VALUES (22, 'DEPT 22');
```

Range Partitioning (8i) (Multi-Column)


```
create table cust_sales (
acct_no    number(5),
cust_name  char(30),
sale_day integer not null,
sale_mth   integer not null,
sale_yr    integer not null)
partition by range (sale_yr, sale_mth, sale_day)
   (partition cust_sales_q1 values less than (1998, 04, 01) tablespace users1,
    partition cust_sales_q2 values less than (1998, 07, 01) tablespace users2,
    partition cust_sales_q3 values less than (1998, 10, 01) tablespace users3,
    partition cust_sales_q4 values less than (1999, 01, 01) tablespace users4,
    partition cust_sales_qx values less than (maxvalue, maxvalue, maxvalue)
     tablespace users4);
```

Hash Partitioning (8i) (Multi-Column)


```
create table cust_sales_hash (
acct_no number(5),
cust_name  char(30),
sale_day integer not null,
sale_mth  integer not null,
sale_yr  integer not null)
partition by hash (acct_no)
partitions 4
store in (users1, users2, users3, users4);
```

Composite Partitioning v (8i)

```
CREATE TABLE test5 (data_item INTEGER, length_of_item INTEGER,
        storage_type VARCHAR(30),                    owning_dept NUMBER,
   storage_date DATE)   PARTITION BY RANGE (storage_date)   SUBPARTITION BY
   HASH(data_item)   SUBPARTITIONS 4
    STORE IN (data_tbs1, data_tbs2,
              data_tbs3, data_tbs4)   (PARTITION q1_1999      VALUES LESS
   THAN (TO_DATE('01-apr-1999', 'dd-mon-yyyy')),    PARTITION q2_1999
   VALUES LESS THAN (TO_DATE('01-jul-1999', 'dd-mon-yyyy')),
     PARTITION q3_1999
       VALUES LESS THAN (TO_DATE('01-oct-1999', 'dd-mon-yyyy'))
       (SUBPARTITION q3_1999_s1 TABLESPACE data_tbs1,
        SUBPARTITION q3_1999_s2 TABLESPACE data_tbs2),
     PARTITION q4_1999
       VALUES LESS THAN (TO_DATE('01-jan-2000', 'dd-mon-yyyy'))
       SUBPARTITIONS 8
       STORE IN (q4_tbs1, q4_tbs2, q4_tbs3, q4_tbs4,
                 q4_tbs5, q4_tbs6, q4_tbs7, q4_tbs8),    PARTITION q1_2000
      VALUES LESS THAN (TO_DATE('01-apr-2000', 'dd-mon-yyyy')));
```

List Partitioning (Allowed since 9i)


```
create table dept_part
(deptno     number(2),
dname       varchar2(14),
loc         varchar2(13))
partition by list (dname)
(partition d1_east  values ('BOSTON', 'NEW YORK'),
partition d2_west  values ('SAN FRANCISCO', 'LOS ANGELES'),
partition d3_south values ('ATLANTA', 'DALLAS'),
partition d4_north values ('CHICAGO', 'DETROIT'));

Table created.
```

Interval Partitioning – 11g

- This is a helpful addition to range partitioning where Oracle automatically creates a partition when the inserted value exceeds all other partition ranges. **11g also has Ref & Virtual Column Partitioning (not covered here).**

There are the following restrictions:

- You can only specify one partitioning key column, and it must be of NUMBER or DATE type.
- Interval partitioning is NOT supported for index-organized tables.
- Interval Partitioning supports composite partitioning:
  - Interval-range *** Interval-hash *** Interval-list
- You can NOT create a domain index on an interval-partitioned table.

Interval Partitioning – 11g

```
CREATE TABLE DEPT_new
(DEPTNO         NUMBER(2),
 DEPT_NAME  VARCHAR2(30))
 PARTITION BY RANGE(DEPTNO)
      (PARTITION D1 VALUES   LESS THAN (10),
       PARTITION D2 VALUES     LESS THAN (20),
       PARTITION D3 VALUES     LESS THAN (30));

Table created.

SQL> insert into dept_new values(40, 'test2');
insert into dept_new values(40, 'test2')
            *
ERROR at line 1:
ORA-14400: inserted partition key does not map to any partition
```

Interval Partitioning – 11g

```
select segment_name, partition_name
from dba_segments
where segment_name = 'DEPT_NEW';
```

```
SEGMENT_NAME                   PARTITION_NAME
------------------------------ ------------------------------
DEPT_NEW                       D1
DEPT_NEW                       D2
DEPT_NEW                       D3
```

Interval Partitioning – 11g

```
CREATE TABLE DEPT_NEW2
(DEPTNO        NUMBER(2),
 DEPT_NAME  VARCHAR2(30))
 PARTITION BY RANGE(DEPTNO)
 INTERVAL(10)
      (PARTITION D1 VALUES   LESS THAN (10),
       PARTITION D2 VALUES     LESS THAN (20),
       PARTITION D3 VALUES     LESS THAN (30))
```

Table created.

```
SQL> insert into dept_new2 values(40, 'test2');
1 row created.
```

Interval Partitioning – 11g

```
insert into dept_new2 values(40,null);
insert into dept_new2 values(50,null);
insert into dept_new2 values(99,null);

select  segment_name, partition_name
from   dba_segments
where  segment_name = 'DEPT_NEW2'

SEGMENT_NAME     PARTITION_NAME
---------------------------- ------------------------------
DEPT_NEW2            D1
DEPT_NEW2            D2
DEPT_NEW2            D3
DEPT_NEW2            SYS_P41
DEPT_NEW2            SYS_P42
DEPT_NEW2            SYS_P43
```

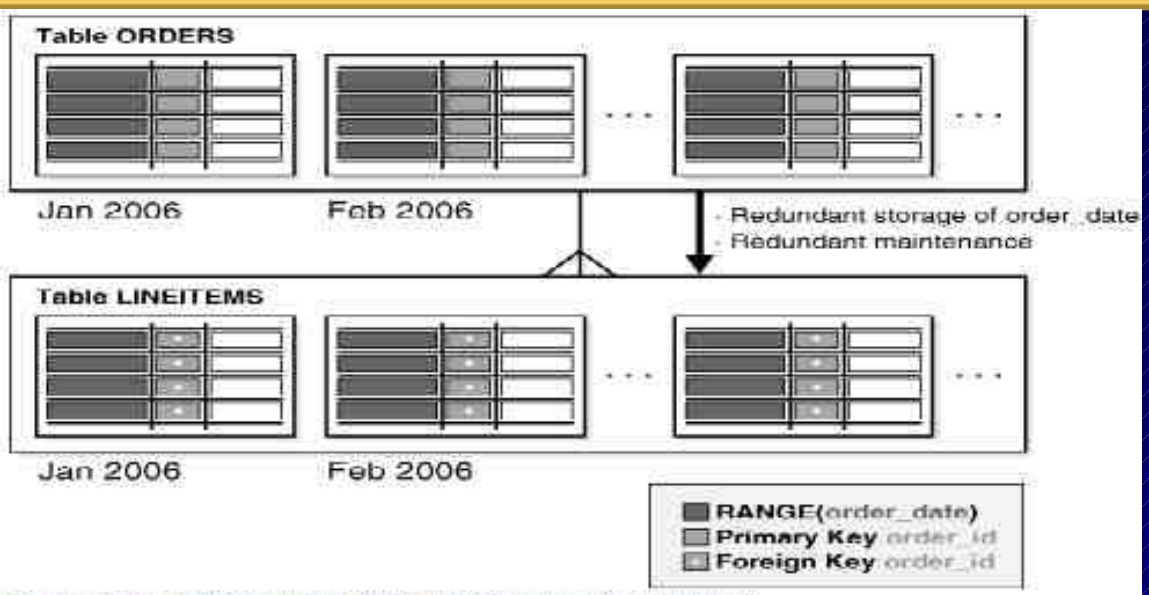
Reference Partitioning – 11g (FYI Only)

- Allows the partitioning of two tables related to one another by referential constraints. The partitioning key is resolved through an existing parent-child relationship, enforced by enabled and active primary key and foreign key constraints.
- Tables with a parent-child relationship can be logically equi-partitioned by inheriting the partitioning key from the parent table without duplicating the key columns. The logical dependency will also automatically cascade partition maintenance operations, thus making application development easier and less error-prone.

Reference Partitioning – 11g (FYI Only)

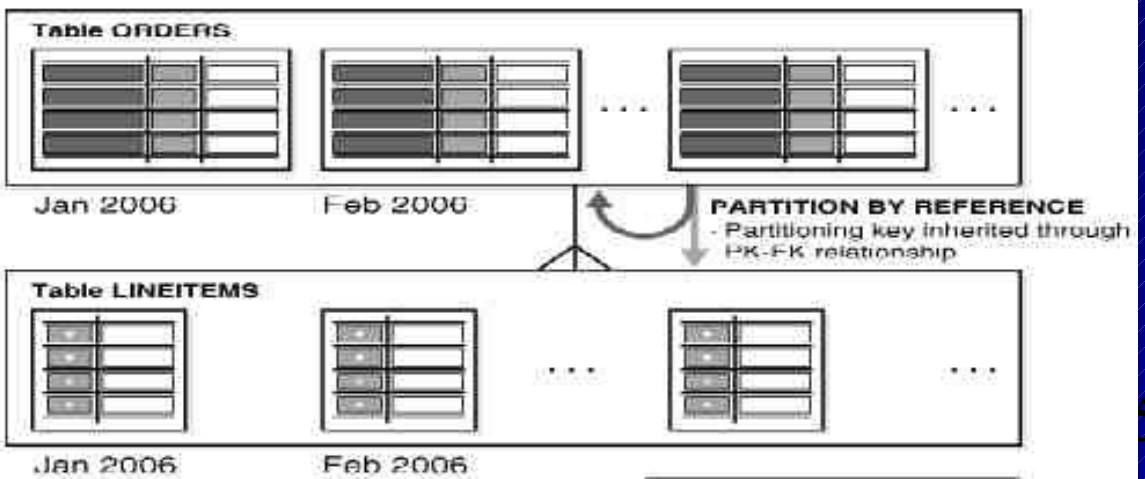
```
CREATE TABLE orders
( order_id      NUMBER(12)    CONSTRAINT orders_order_id_nn    NOT NULL,
  order_date    DATE          CONSTRAINT orders_order_date_nn  NOT NULL,
  order_mode    VARCHAR2(8),
  customer_id   NUMBER(6)     CONSTRAINT orders_customer_id_nn NOT NULL,
  order_status  VARCHAR2(2),
  order_total   NUMBER(8,2),
  sales_rep_id  NUMBER(6),
  promotion_id  NUMBER(6),
 CONSTRAINT orders_order_id_pk PRIMARY KEY (order_id)
)
PARTITION BY RANGE (order_date)
  ( PARTITION p_pre_1999 VALUES LESS THAN (TO_DATE('01-JAN-2006','dd-MON-yyyy')),
   PARTITION p_JAN_1999 VALUES LESS THAN (TO_DATE('01-FEB-2006','dd-MON-yyyy'))
)
PARALLEL ;

CREATE TABLE lineitems
( order_id      NUMBER(12)    CONSTRAINT oitems_order_id_nn      NOT NULL,
  line_item_id  NUMBER(3)     CONSTRAINT oitems_line_item_id_nn  NOT NULL,
  product_id    NUMBER(6)     CONSTRAINT oitems_product_id_nn    NOT NULL,
  unit_price    NUMBER(8)     CONSTRAINT oitems_unit_price_nn    NOT NULL,
  quantity      NUMBER(8,2)   CONSTRAINT oitems_quantity_nn      NOT NULL,
  sales_amount  NUMBER(12,2)  CONSTRAINT oitems_sales_amount_nn  NOT NULL,
  CONSTRAINT order_items_orders_fk
  FOREIGN KEY (order_id) REFERENCES orders(order_id)
)
PARTITION BY REFERENCE (order_items_orders_fk)
PARALLEL ;
```

Reference Partitioning – 11g



Description of "Figure 2-4 Before Reference Partitioning"

Figure 2-5 With Reference Partitioning



Partition Compression

- You can now COMPRESS individual partitions
- Compression as high as 3.5 to 1 is possible
- Compressed Tables now support
  - DML Statements
  - Add and Drop Column
  - Partition level COMPRESS or NOCOMPRESS
- ALTER TABLE... COMPRESS (old compress)
- ALTER TABLE... NOCOMPRESS
- Table compression now supported for OLTP
- New Advanced Compression Option (chargeable):
  - CREATE TABLE *t1* COMPRESS FOR ALL OPERATIONS

Partition Compression

```
CREATE TABLE DEPT_new3
   (DEPTNO            NUMBER(2),
    DEPT_NAME         VARCHAR2(30))
    COMPRESS
    PARTITION BY RANGE(DEPTNO)
    interval(10)
    (PARTITION D1 VALUES   LESS THAN (10),
     PARTITION D2 VALUES    LESS THAN (20) NOCOMPRESS,
     PARTITION D3 VALUES    LESS THAN (30))
```

Table created.

Partition Compression

```
insert into dept_new3 values(10,null);
1 row created.
insert into dept_new3 values(20,null);
1 row created.
insert into dept_new3 values(30,null);
1 row created.
insert into dept_new3 values(60,null);
1 row created.
insert into dept_new3 values(90,null);
1 row created.
```

Partition Compression

```
select  table_name, partition_name, compression
from    dba_tab_partitions
where table_name = 'DEPT_NEW3';
```

```
TABLE_NAME                     PARTITION_NAME                 COMPRESS
------------------------------ ------------------------------ ----------
DEPT_NEW3                      D1                             ENABLED
DEPT_NEW3                      D2                             DISABLED
DEPT_NEW3                      D3                             ENABLED
DEPT_NEW3                      SYS_P64                        ENABLED
DEPT_NEW3                      SYS_P65                        ENABLED
DEPT_NEW3                      SYS_P66                        ENABLED
```

6 rows selected.

Advanced Compression

Compression History – Timeline (FYI Only)

- Index Compression since 8i
- Table Compression since 9i
  - No Additional License Requirement
  - Only for direct inserts
  - Compression Not Maintained with updates and normal inserts
  - Had to re-org table to re-compress over time.
- 11g Advanced Compression
  - Additional License Requirement
  - Compression Maintained with all DML activity
  - No re-orgs required after initial compression
- 11gR2 – Hybrid Columnar Compression (with Exadata)

Advanced Compression (FYI Only)

- The Oracle Advanced Compression option contains the following features:
  - Data Guard Network Compression
  - Data Pump Compression (COMPRESSION=METADATA_ONLY does not require the Advanced Compression option)
  - Multiple RMAN Compression Levels (RMAN DEFAULT COMPRESS does not require the Advanced Compression option)
  - OLTP Table Compression
  - SecureFiles Compression and Deduplication. LZO compression algorithm added. Faster than ZLIB.

Advanced Compression – Test Case (FYI Only)

- A test Example – basic CTAS clause:

```
select mysequence.nextval seq_id, owner, table_name,
   tablespace_name, cluster_name, iot_name,
   dbms_random.string('A',10) random_string
from dba_tables  where 1=2;
```

- Test scenario
  - 2 sets of 2 tables, each with a sequence based PK.
  - One set with a random data value one without.
  - Within a set, one table is compressed and the other is not.
  - Run a Standard PL/SQL block, iterating 100 times to load data.
    - Use dbms_random for data for table with random column information.
    - Use 'XXXXXXXXXX' for the second set.

Advanced Compression – Results

Table Sizes and storage

Table Name	Kbytes	Ratio
ACTEST_NORANDOM_COMP	11776	**36.11%**
ACTEST_NORANDOM_NORMAL	18432	
ACTEST_RANDOM_COMP	14848	**17.14%**
ACTEST_RANDOM_NORMAL	17920	

Data Load Time Comparisons

Table Name	Seconds	Ratio
ACTEST_NORANDOM_COMP	22.58	**60.03%**
ACTEST_NORANDOM_NORMAL	14.11	
ACTEST_RANDOM_COMP	26.28	**21.78%**
ACTEST_RANDOM_NORMAL	21.58	

Advanced Compression & OLTP

- Compression is maintained at a block level.
- Maintained through DML operations.
- Compression ratio depends on "RANDOMness" of the data.
- DML Impact depends on "RANDOMness" of the data. Range of 10 to 30%.
  - More visible for bulk operations compared to single row operations.
- Significant Performance gains in selects, primarily due to reduced block scans.
  - Exacts / specifics depend on your compression ratio.
  - Table scans expect upto 50% reduction in block reads.
  - PK based access, impact not noticeable.
- Must evaluate on a case by case basis.

Audience Knowledge

Exadata V1?

10x faster than any Oracle DW

Exadata V2? X2-8?

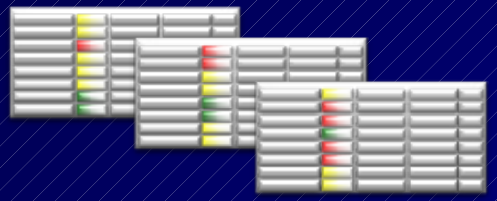
5x faster than V1

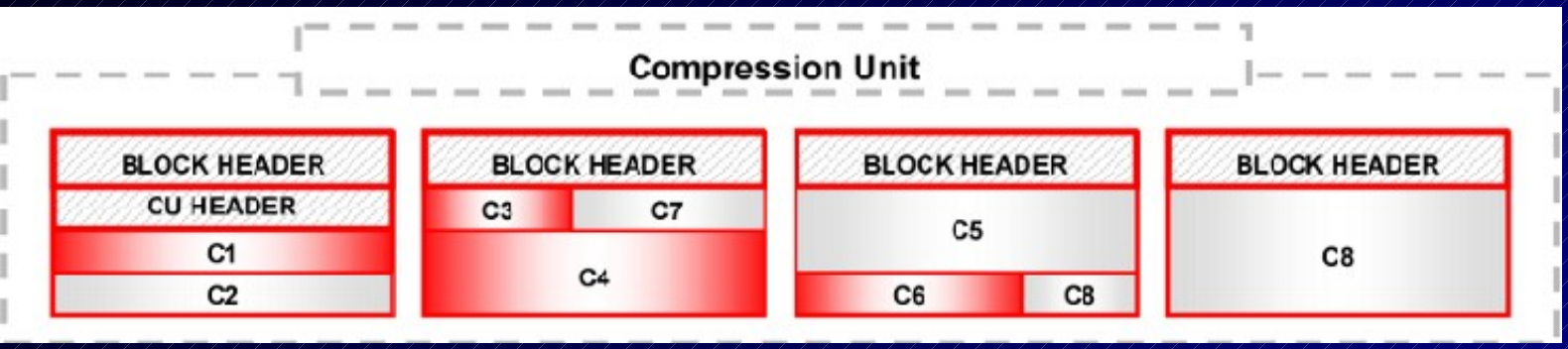
Exadata Hybrid Columnar Compression (EHCC) – 4-10x & 30x is common

- What is it (a HYBRID of column & row storage)?
  - Data organized by column and compressed vs. row
  - Tables organized in Compression Units (CU)-1000 rows?
  - CU's span many blocks (32K)
  - Good for data bulk loaded (not for OLTP – single block)
- What's it for?
  - Query Data / DWHS (**NOT frequently Updated**)
- How much does it compress (old OLTP was 2-3x)?
  - 10x in a typical data warehouse compression; (*we got 4-11)*
  - 15x to 70x in archive compression (*cold* data); (*we got 32)*

Hybrid Columnar Compression

1. Column Data Compressed





(Warehouse)

2. Stored in Compression Units
(Better compression when column data stored together)

(Archive)

*** Thanks Oracle for these images*

Hybrid Columnar Compression

- Faster Operations: Query runs without decompression
  - Compressed/Processed in FLASH CACHE; lower I/O!
  - Compressed when sent over InfiniBand!
  - Cloned compressed!
  - Backed Up compressed!
  - Scans MUCH less (compressed) data
- Worth Noting:
  - Use standard table compression for OLTP
  - Single block lookup FASTER than other columnar storage
  - Updated rows migrate to normal / lower level compression

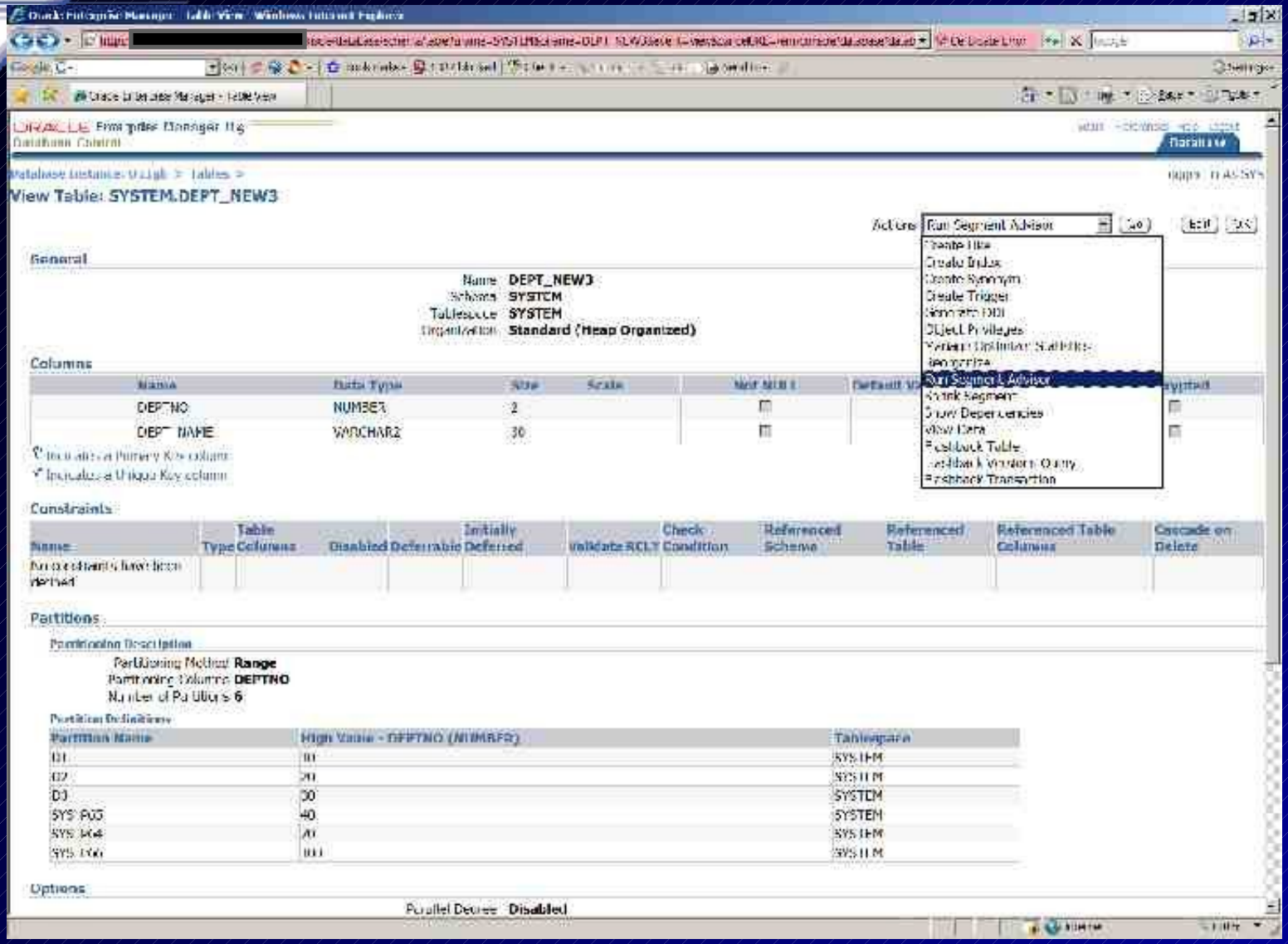
Hybrid Columnar Compression

- Fully supported:
  - B-Tree Indexes
  - Bitmap Indexes
  - Text Index
  - Materialized Views
  - Partitioning
  - Parallel Query
  - Data Guard Physical Standby
  - *Logical Standby and Streams (FUTURE release)*
  - ***Smart Scans of HCC tables!***

Other Oracle Compression

- Data Pump Compression
  - Compression = {ALL | DATA_ONLY | NONE}
- RMAN Backup Compression
  - Compression Level LOW/HIGH (New in 11.2)
- Secure File Compression
  - LOW/MEDIUM/HIGH (2-3x compression)
  - Deduplication & Encryption
- Normal OLTP Table Compression (since 9.2)
  - 11g now supports INSERT/UPDATE
  - FASTER Algorithm
- Data Guard Redo Transport Compression

Object Maintenance



Object Maintenance – Reorganize

Automated Maintenance Tasks

- Automatic Optimizer Statistics Collection
- Automatic Segment Advisor
- Automatic SQL Tuning Advisor (DBMS_SQLTUNE)
- Disable/Enable Automated Tasks:

DBMS_AUTO_TASK_ADMIN.DISABLE (ENABLE)

- Setting up Maintenance Windows

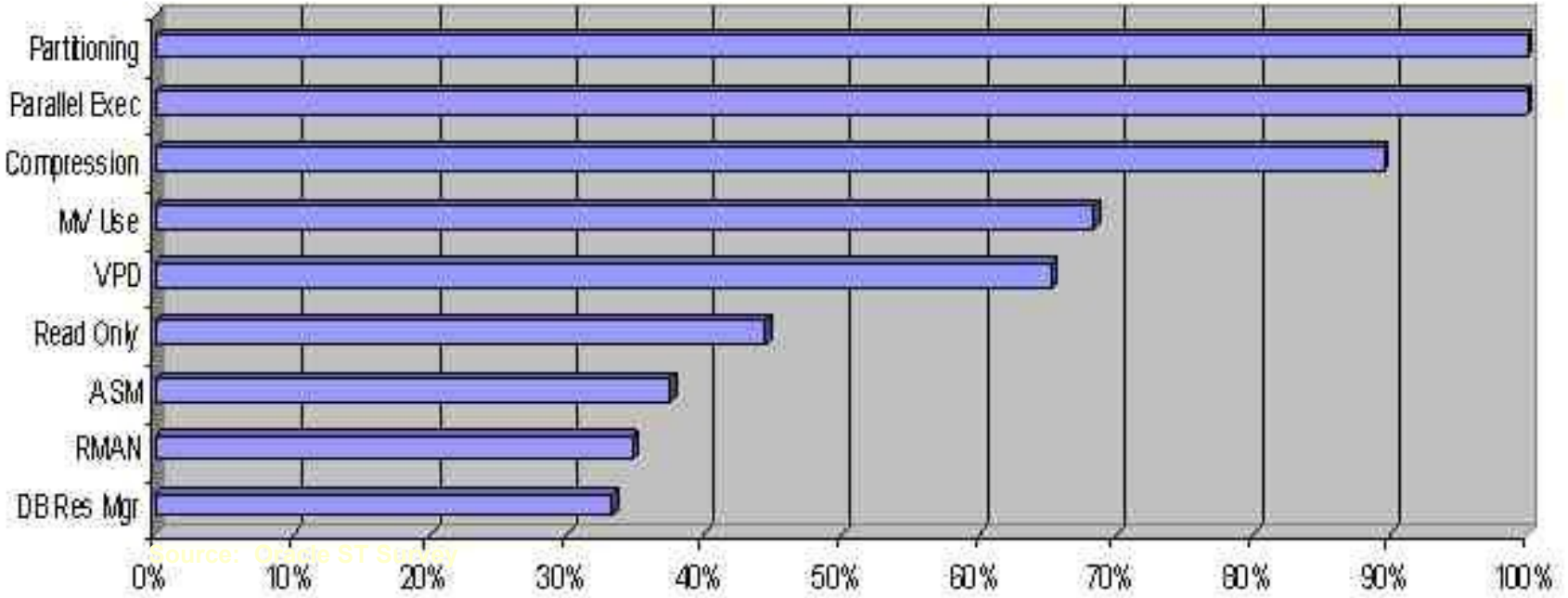
DBMS_SCHEDULER.CREATE_WINDOW

Additional Enhancements

- Ability to online redefine tables that have materialized view logs:
  - Tables with materialized view logs can now be redefined online.
  - Materialized view logs are now one of the dependent objects that can be copied to the interim table with the DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS package procedure.
- DBMS_STATS performance has been improved.

Large-Scale Data Warehouses*

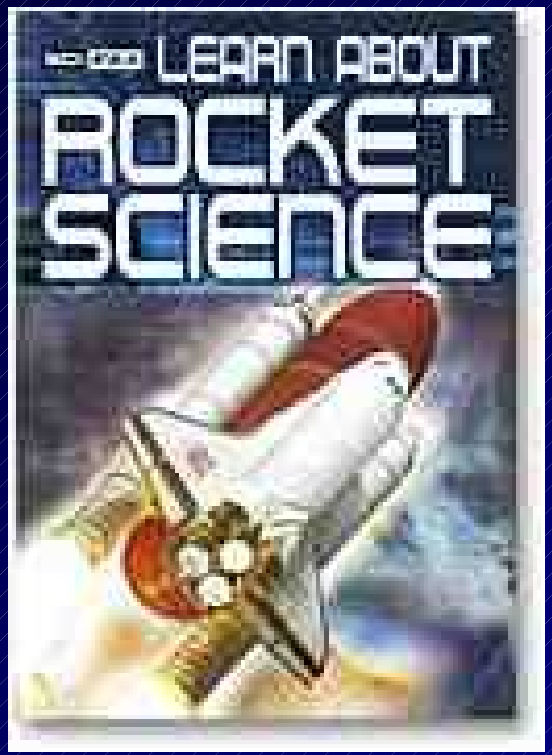
Feature Usage



Source: Oracle ST Survey

* Oracle Survey

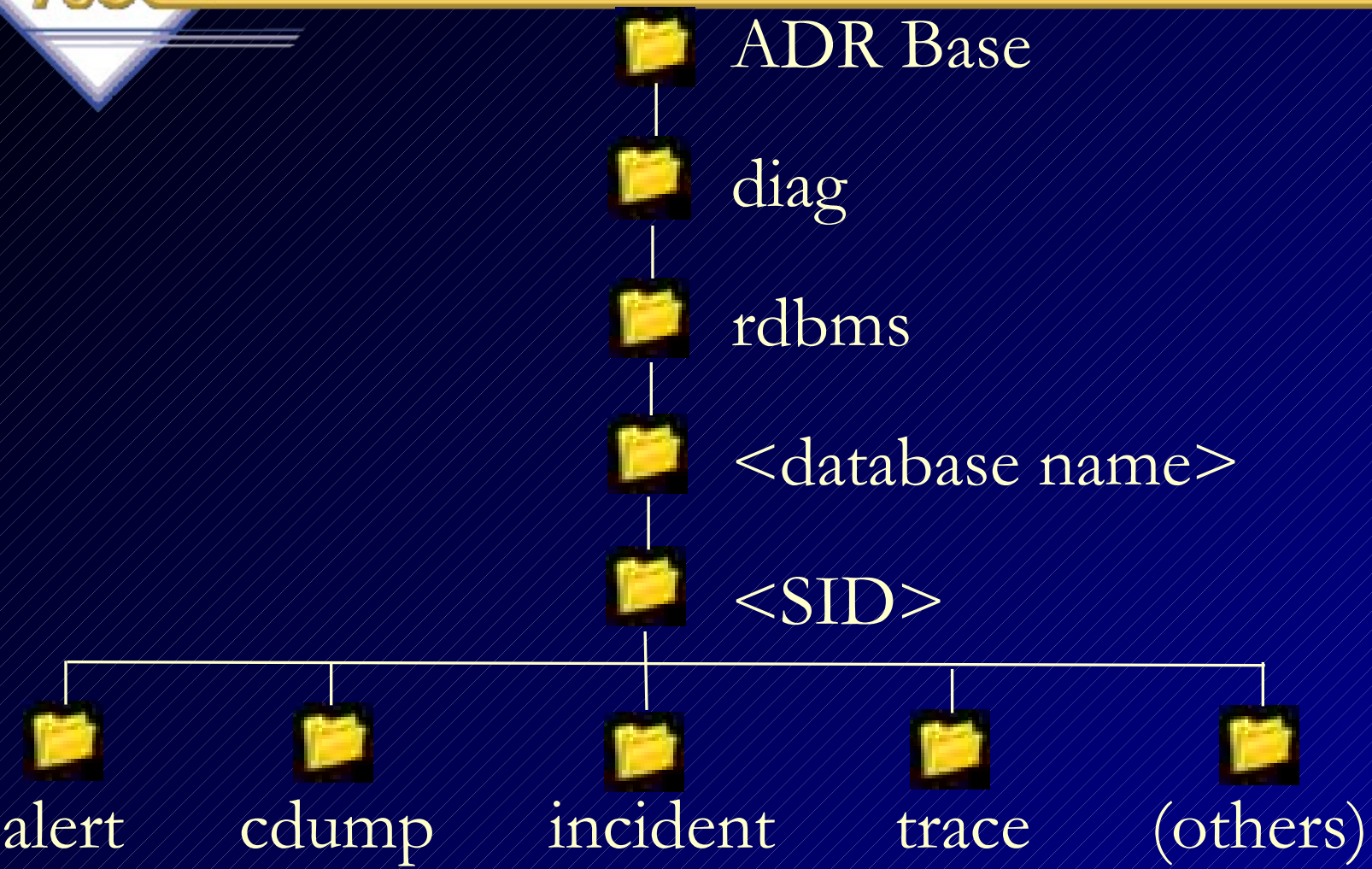
Automatic Diagnostic Repository (ADR)



Automatic Diagnostic Repository (ADR)

- Oracle 11g includes a Fault Diagnosability Infrastructure to prevent, detect, diagnose, resolve issues related to bugs, corruption, etc.
- When a critical error occurs it is assigned an incident number and all diagnostic data tagged with this in ADR.
- ADR is a file based repository outside of the database
- ADR helps detect problems proactively
- ADR helps limit the damage of interruptions
- ADR helps reduce problem diagnostic time
- ADR simplifies Oracle Support / Customer interaction
- The ADR also contains Health Reports, Trace Files, Dump Files, SQL Test Cases and Data Repair Records

ADR Directory Structure for a Database Instance



Alert Log: /u01/app/oracle/diag/rdbms/o11gb/O11gb/trace
ORACLE_HOME: /u01/app/oracle/product/11.1.0/db_1

ADR – V$ Diagnostic Info

select name, value
from v$diag_info;

```
NAME                           VALUE
------------------------------ ----------------------------------------------------
Diag Enabled                   TRUE
ADR Base                       /u01/app/oracle
ADR Home                       /u01/app/oracle/diag/rdbms/o11gb/O11gb
Diag Trace                     /u01/app/oracle/diag/rdbms/o11gb/O11gb/trace
Diag Alert                     /u01/app/oracle/diag/rdbms/o11gb/O11gb/alert
Diag Incident                  /u01/app/oracle/diag/rdbms/o11gb/O11gb/incident
Diag Cdump                     /u01/app/oracle/diag/rdbms/o11gb/O11gb/cdump
Health Monitor                 /u01/app/oracle/diag/rdbms/o11gb/O11gb/hm
Default Trace File             /u01/app/oracle/diag/rdbms/o11gb/O11gb/trace/O11gb_ora_16676.trc
Active Problem Count           0
Active Incident Count          0

11 rows selected.
```

ADR – V$ Diagnostic Info
11R2 – No changes (that I saw)

```
  1* select * from V$diag_info
SYS@sillgr2> /

   INST_ID NAME                           VALUE
---------- ------------------------------ --------------------------------------------------------------------------------
         1 Diag Enabled                   TRUE
         1 ADR Base                       /u01/app/oracle
         1 ADR Home                       /u01/app/oracle/diag/rdbms/sillgr2/sillgr2
         1 Diag Trace                     /u01/app/oracle/diag/rdbms/sillgr2/sillgr2/trace
         1 Diag Alert                     /u01/app/oracle/diag/rdbms/sillgr2/sillgr2/alert
         1 Diag Incident                  /u01/app/oracle/diag/rdbms/sillgr2/sillgr2/incident
         1 Diag Cdump                     /u01/app/oracle/diag/rdbms/sillgr2/sillgr2/cdump
         1 Health Monitor                 /u01/app/oracle/diag/rdbms/sillgr2/sillgr2/hm
         1 Default Trace File             /u01/app/oracle/diag/rdbms/sillgr2/sillgr2/trace/sillgr2_ora_8978.trc
         1 Active Problem Count           1
         1 Active Incident Count          2

11 rows selected.

SYS@sillgr2>
SYS@sillgr2>
SYS@sillgr2>
```

Optimizer Statistics & Other Optimizer Advances

Special Thanks: Maria Colgan, Penny Avril & Debbie Migliore

Improved SPEED and Quality Gathering Stats – AUTO-SAMPLING

- Manually gather stats: Impossible to find sample size that works for ALL tables - need COMPUTE
- Especially hard to find a good sample size when the data distribution is very skewed.
- NEW Auto-sampling: "Discovers" the best sample size for every table in your system for you.
  - Get the Quality of a COMPUTE with SPEED of a SAMPLE
  - Oracle' goal is to OBSOLETE the need and use of sampling.
  - Accuracy is comparable to COMPUTE

Incremental Statistics Maintenance - Stats by Partition vs. table

- In 10g, if you gather stats on one partition **after a bulk load** it causes a full scan of all partitions to gather global table statistics with is extremely time consuming
- In 10g, you have to manual copy statistics to new partition
- In 11g Gather stats for TOUCHED PARTITIONS only!
- Table stats are refreshed WITHOUT scanning the un-touched partitions.

Manage New Statistics
Gather Stats but make PENDING

- Currently DBAs are scared to gather stats on a table that is changing for fear of unpredictable execution plans.
- You have to 'FREEZE' critical plans or stats.
- In 11g, gather stats and save as PENDING.
- Verify the new stats won't adversely affect things by checking them with a single user using an alter session or try them out on a different system.
- When everything looks good – then, PUBLISH them for all to use!

Manage New Statistics
Gather Stats but make them PENDING


```
select dbms_stats.get_prefs('PUBLISH', 'SH', 'CUST') publish from dual;

PUBLISH
--------------------
TRUE

exec dbms_stats.set_table_prefs('SH', 'CUST', 'PUBLISH', 'false');
PL/SQL procedure successfully completed.

select dbms_stats.get_prefs('PUBLISH', 'SH', 'CUST') publish from dual;

PUBLISH
--------------------
FALSE
```

Manage New Statistics
Gather Stats but make them PENDING

```
select table_name, last_analyzed analyze_time, num_rows, blocks, avg_row_len
from user_tables
where table_name = 'CUST';

TABLE_NAME ANALYZE_T    NUM_ROWS      BLOCKS AVG_ROW_LEN
---------- ---------- ---------- ---------- ------------
CUST

execute dbms_stats.gather_table_stats('SH', 'CUST');
PL/SQL procedure successfully completed.

select table_name, last_analyzed analyze_time, num_rows, blocks, avg_row_len
from user_tables
where table_name = 'CUST';

TABLE_NAME ANALYZE_T    NUM_ROWS      BLOCKS AVG_ROW_LEN
---------- ---------- ---------- ---------- ------------
CUST
```

Manage New Statistics
PUBLISH Stats after Testing Complete


```
alter session set optimizer_use_pending_statistics = true;
```

(Then run your query – If ready/better – publish the new stats)

```
exec dbms_stats.publish_pending_stats('SH', 'CUST');
PL/SQL procedure successfully completed.

select table_name, last_analyzed analyze_time, num_rows, blocks, avg_row_len
from user_tables
where table_name = 'CUST';

TABLE_NAME ANALYZE_T   NUM_ROWS      BLOCKS AVG_ROW_LEN
---------- --------- ---------- ---------- -----------
CUST       13-OCT-07      55500       1485         180

exec dbms_stats.delete_table_stats('SH', 'CUST'); <to delete>
```

Extended Optimizer Statistics: New Multi-Column Statistics

- Corporate data often has correlations between different columns of a table. For example:
  - A job title is correlated to the salary.
  - The season affects the sold amounts of items such as swim suits sell more in the summer and snow shoes sell more in the winter.
  - The make of a car and color are often used together but are not really correlated well so the filter doesn't reduce the result set.
- Optimizer has to estimate the correct cardinality
  - *Will the additional column condition reduce the result set or not? Should it be used.*
- Oracle calculates correlated statistics so the optimizer will make great decisions. Single column statistics and histograms are not enough!

Example

```
SELECT make, price, color
FROM   cars_dot_com
WHERE make = 'CORVETTE';
```

CORVETTE	40,000	RED
CORVETTE	60,000	BLACK
CORVETTE	50,000	SILVER

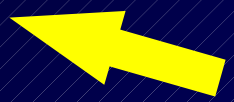
- Three records selected.
- Single column statistics are accurate

Make	Price	Color
CORVETTE	40,000	RED
CORVETTE	60,000	BLACK
CORVETTE	50,000	SILVER
CADILLAC	90,000	RED
JEEP	35,000	BLACK
JEEP	45,000	SLIVER

Example, cont.

```
SELECT make, price, color
FROM   cars_dot_com
WHERE  make = 'CORVETTE'
AND    COLOR = 'RED';
```

CORVETTE	40,000	RED



Make	Price	Color
CORVETTE	40,000	RED
CORVETTE	60,000	BLACK
CORVETTE	50,000	SILVER
CADILLAC	90,000	RED
JEEP	35,000	BLACK
JEEP	45,000	SLIVER

- One record selected.
  - No correlated columns
  - Additional predicate **reduces result set**
  - Single column statistics are STILL sufficient

Example, cont.

```
SELECT make, price, color
FROM   cars_dot_com
WHERE  make = 'CORVETTE'
AND    PRICE = 50000;
```

CORVETTE	50,000	RED
CORVETTE	50,000	BLACK
CORVETTE	50,000	SLIVER

Make	Price	Color
CORVETTE	50,000	RED
CORVETTE	50,000	BLACK
CORVETTE	50,000	SILVER
CADILLAC	90,000	RED
JEEP	35,000	BLACK
JEEP	45,000	SLIVER

- Three records selected.
  - Correlated columns
  - Additional predicate has no effect
  - **Single column statistics are NOT sufficient**
  - **Must use '=' and not < or >**

Manage New Statistics – FYI Only
EXTENDED Statistic Group

- Provides a way to collect stats on a group of columns
- Full integration into existing statistics framework
  - Automatically maintained with column statistics
  - Instantaneous and transparent benefit for any application
- Accurate cardinalities for inter-related columns
  - Multiple predicates on the same table are estimated correctly

Manage New Statistics – FYI Only
After normal Statistics Creation

select column_name, num_distinct, histogram
from user_tab_col_statistics where table_name = 'CUSTOMERS';

```
COLUMN_NAME                    NUM_DISTINCT HISTOGRAM
------------------------------ ------------ ---------------
CUST_VALID                                2 NONE
COUNTRY_ID                               19 FREQUENCY
CUST_STATE_PROVINCE                     145 NONE
CUST_CITY_ID                            620 HEIGHT BALANCED
CUST_CITY                               620 NONE
CUST_LAST_NAME                          908 NONE
CUST_FIRST_NAME                        1300 NONE
CUST_ID                               55500 NONE
...
23 rows selected.
```

Manage New Statistics – FYI Only
Create EXTENDED Statistic Group

- Now lets create the extended statistics group & re-gather statistics on the CUSTOMER table (query user_tab_col_statistics to see new column):

```
select  dbms_stats.create_extended_stats('SH','CUSTOMERS', '(country_id,
    cust_state_province)') from dual;

DBMS_STATS.CREATE_EXTENDED_STATS('SH','CUSTOMERS','(CO
------------------------------------------------------------------------
SYS_STUJGVLRVH5USVDU$XNV4_IR#4

exec dbms_stats.gather_table_stats('SH','CUSTOMERS', method_opt =>
    'for all columns size skewonly');
PL/SQL procedure successfully completed.
```

Manage New Statistics – FYI Only
Now there are Extended Statistics

select column_name, num_distinct, histogram
from user_tab_col_statistics where table_name = 'CUSTOMERS';

```
COLUMN_NAME                    NUM_DISTINCT HISTOGRAM
------------------------------ ------------ ---------------
SYS_STUJGVLRVH5USVDU$XNV4_IR#4          145 FREQUENCY
CUST_VALID                                2 FREQUENCY
COUNTRY_ID                               19 FREQUENCY
CUST_STATE_PROVINCE                     145 FREQUENCY
CUST_CITY_ID                            620 HEIGHT BALANCED
CUST_CITY                               620 HEIGHT BALANCED
CUST_LAST_NAME                          908 HEIGHT BALANCED
CUST_FIRST_NAME                        1300 HEIGHT BALANCED
CUST_ID                               55500 HEIGHT BALANCED
...

24 rows selected.
```

Manage New Statistics – FYI Only
DROP Extended Statistics


```
exec dbms_stats.drop_extended_stats('SH', 'CUSTOMERS', '(country_id, cust_state_province)');
PL/SQL procedure successfully completed.

select column_name, num_distinct, histogram
from user_tab_col_statistics where table_name = 'CUSTOMERS';

COLUMN_NAME                    NUM_DISTINCT HISTOGRAM
------------------------------ ------------ ---------------
CUST_VALID                                2 NONE
COUNTRY_ID                               19 FREQUENCY
CUST_STATE_PROVINCE                     145 NONE
CUST_CITY_ID                            620 HEIGHT BALANCED
CUST_CITY                               620 NONE
CUST_LAST_NAME                          908 NONE
CUST_FIRST_NAME                        1300 NONE
CUST_ID                               55500 NONE
...
23 rows selected.
```

Adaptive Cursor Sharing

- The optimizer peeks at user-defined bind values during plan selection on the hard parse.
- Initial value of the binds determines the plan for all future binds (hopefully the first peek covers most queries)
- Same execution plan shared regardless of future bind values
- One plan is not always appropriate for all bind values for a given SQL statement
  - Where job= 'PRESIDENT' (use an index – only one row)
  - Where job = 'OPERATOR' (don't use an index – 90% of the table)
- If Oracle "peeks" and sees the President, it will use the index. Future queries also use the index without peeking after that (bad for the OPERATOR query).

Bind Peeking – Pre-11g

- If you need to tune a query that you suspect has issues related to bind peeking, use v$sql_plan or tkprof output using different values for bind variables and compare execution plans in both cases.

- If you wish to deactivate bind peeking you can set:

```
alter system set "_OPTIM_PEEK_USER_BINDS"=FALSE;
```

Note: When running tkprof "explain=username/password" argument should NOT be used. That will cause tkprof to issue an explain plan whose output could differ from the execution plan info inside the raw 10046/sql_trace file.

Consider a Telephone Company…

```
SELECT Ename, Empno, Job
FROM   Emp
WHERE Job = :B1
Value of B1 = 'OPERATOR';
```

Ename	Empno	Job
SMITH	6973	OPERATOR
ALLEN	7499	OPERATOR
WARD	7521	OPERATOR
KING	8739	PRESIDENT
SCOTT	7788	OPERATOR
CLARK	7782	OPERATOR

Ename	Empno	Job
SMITH	6973	OPERATOR
ALLEN	7499	OPERATOR
WARD	7521	OPERATOR
SCOTT	7788	OPERATOR
CLARK	7782	OPERATOR

- If 'OPERATOR' is the bind value at hard parse, most records will be selected. Execution plan will be a full table scan
- If 'PRESIDENT' is the bind value at hard parse, few records will be selected. Execution plan will be an index search

Adaptive Cursor Sharing

Solution:

- In 11g, Oracle uses bind-aware cursor matching.
- Share the plan when binds values are "equivalent"
  - Plans are marked with selectivity range
  - If current bind values fall within range they use the same plan
- Create a new plan if binds are not equivalent
  - Generating a new plan with a different selectivity range

Bind Peeking
Cursor Sharing (cs) Statistics

select sql_id, peeked, executions, rows_processed, cpu_time
from v$sql_cs_statistics; (using the peeked value on the 2nd+ execution)

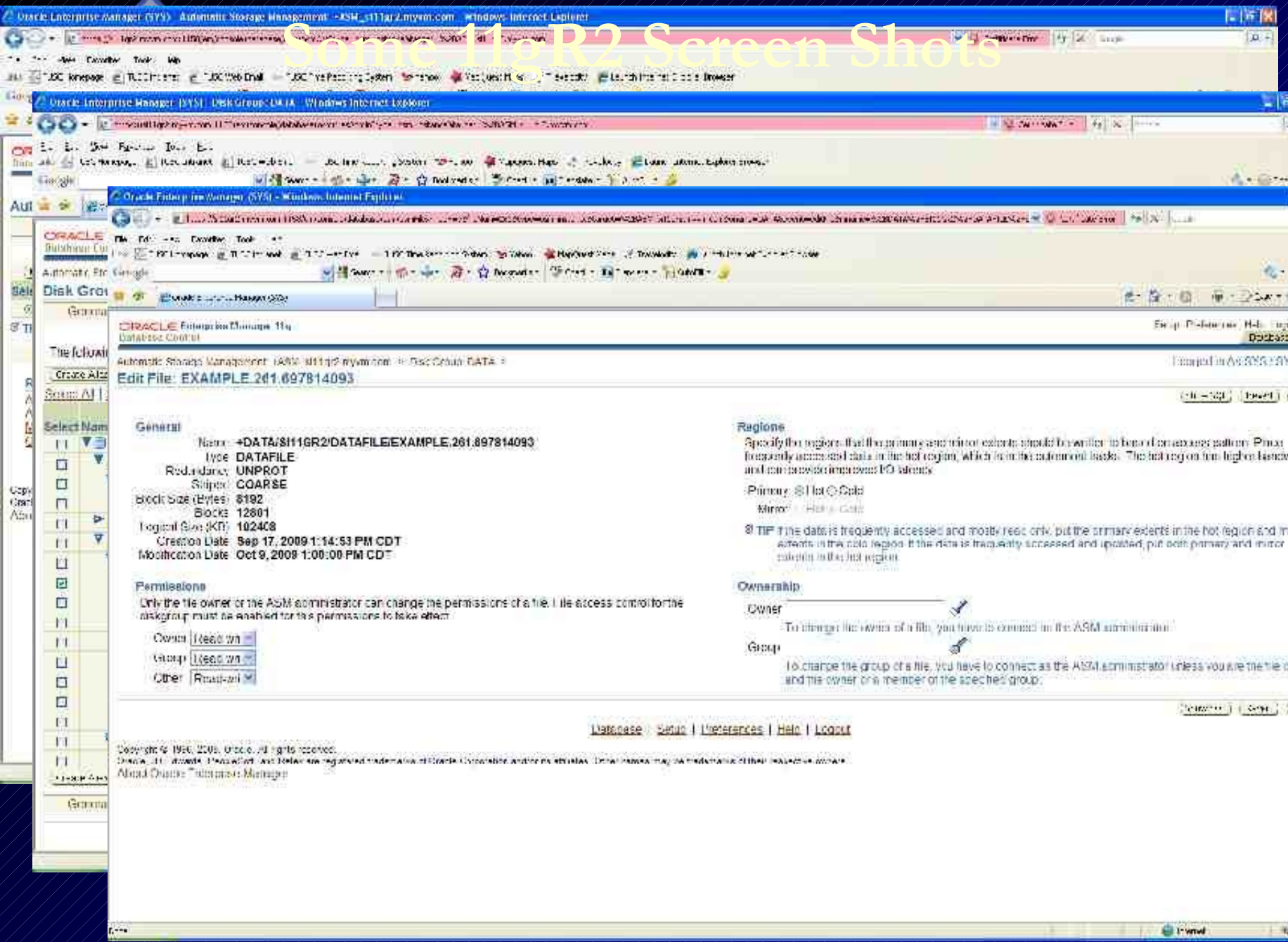
SQL_ID	P	EXECUTIONS	ROWS_PROCESSED	CPU_TIME
5wfj3qs71nd7m	Y	3	1	0
2rad83pp613m1	Y	3	3	0
dr78c03uv97bp	N	1	3	0
dr78c03uv97bp	N	1	3	0
dr78c03uv97bp	Y	1	3	0
9qv6tq9ag5b80	Y	3	3	0
a2k4qkh681fzx	Y	3	2	0
413zr99jf9h72	N	1	1	0
413zr99jf9h72	N	1	1	0
413zr99jf9h72	Y	1	1	0
fd69nfzww1mhm	Y	6	0	0

Bind Peeking – V$SQL

```
select sql_id, executions, is_bind_sensitive, is_bind_aware
from  v$sql;

SQL_ID        EXECUTIONS I I
------------- ---------- - -
9ugwm6xmvw06u          11 Y N
bdfrydpbzw07g          11 Y N
57pfs5p8xc07w          20 N N
...
```

- is_bind_sensitive – If 'Y', then Oracle is using multiple plans depending on bind variable.
- is_bind_aware – Oracle knows that the different data patterns may result depending on bind value. Oracle switches to a bind-aware cursor and may hard parse the statement.

Some 11gR2 Screen Shots



Security Enhancements

Security Enhancements

- 11g is more restrictive
  - Password lock time (1), password grace time (7) and password life time (180) all more restrictive; Failed login attempts stays the same (10).
  - Passwords will be case sensitive now! (on by default)
  - Enhanced hashing algorithm for passwords / DES still available.
  - Strong passwords (set via password complexity verification in EM or SQL):
    - Minimum 8 characters
    - At least one letter and one digit
    - Not servername or servername(1-100)
    - Not a common password (i.e. welcome1)
    - Must differ from previous password by 3 characters minimum

Security Enhancements
AUDIT_TRAIL=DB (default)

- Audit Trail is ON by default (was off in 10g),
- AUDIT_TRAIL=DB is now the default.
- Things that will be audited by default include:
  - CREATE USER, CREATE SESSION, CREATE ANY TABLLE, CREATE ANY PROCEDURE, CREATE ANY JOB, CREATE EXTERNAL JOB, CREATE ANY LIBRARY, CREATE PUBLIC DB LINK
  - ALTER USER, ALTER ANY TABLE, ALTER ANY PROCEDURE, ALTER PROFILE, ALTER DATABASE, ALTER SYSTEM, AUDIT SYSTEM
  - DROP USER, DROP ANY TABLE, DROP ANY PROCEDURE, DROP PROFILE
  - GRANT ANY PRIVILEGE, GRANT ANY OBJECT PRIVILEGE
  - EXEMPT ACCESS POLICY
  - AUDIT SYSTEM
- Cost of Auditing improved to be 1-2% cost on TPCC benchmark.

All the Rest worth noting…

- SEC_CASE_SENSITIVE_LOGON=FALSE
- CONNECT Role only Create Session (vs. Tbl/View…)
- Consider: _NEW_INITIAL_JOIN_ORDERS=FALSE

(CBO more join orders – higher parse times possible)

- GATHER_STATS_JOB on for all DML:

DBMS_STATS.LOCK_TABLE_STATS('SH','T1');

- Auto PROFILES if 3x better; Oracle Always Tuning…
- Statspack STILL works in 11g
- Real Time stats generated for high cpu queries – Careful!
- Generate System Stats on migrate: Tune / 11g Parameters

Oracle Database Security
Built over MANY years...

ORACLE 11g DATABASE

Oracle Audit Vault

Oracle Database Vault

DB Security Evaluation #19

Transparent Data Encryption

EM Configuration Scanning

Fine Grained Auditing (9i)

Secure application roles

Client Identifier / Identity propagation

Oracle Label Security (2000)

Proxy authentication

Enterprise User Security

Global roles

Virtual Private Database (8i)

Database Encryption API

Strong authentication (PKI, Kerberos, RADIUS)

Native Network Encryption (Oracle7)

Database Auditing

Government customer

1977

2007

Other 11gR2 Features

- Grid Plug and Play!!
- Oracle Restart – DB, ASM, Listener after restart of software/hardware
- Out of Place Upgrades (zero downtime for patching)
- In Memory Parallel Execution & Auto Degree of Parallelism (DOP)
- Enterprise Manager for Provisioning, Clusterware, GPnP, Restart
- Universal installer (Remove RAC, de-install, downgrades, patches,restarts)
- ASM FS (file system) snapshots – 64 images – backup/reco/data mining!
- Intelligent data placement on fast tracks
- Flashback Data Archive support for DDL
- Instance caging – allocate CPU usage to instances (CPU_COUNT)
- Compare SQL Tuning sets to each other
- Tuning Advisor can use auto DOP, searches historical performance, transport back to 10gR2 or later for testing.
- Virtual Columns can be in PK/FK of reference partition table
- Stored outline migration to SQL Plan Management (SPM)
- Automatic Block Repair

Other 11gR2 Features

- Exadata simulation – identify areas that benefit from better I/O interconnect throughput
- Backup to Amazon S3 – Simple Storage Solution – Using OSB (Oracle Secure Backup Cloud Computing)
- Email notifications on all 11gR2 jobs
- Tablespace Point in Time Recovery
- Runtime only install of Apex
- Table annotations for Result Cache support (whether to use c/s)
- Oracle Spacial supports truck routing data sets to produce directions that include restrictions on truck use roads, weight, height, time of day, conditions.

Oracle Upgrade Case Studies

(Thanks Mike Dietrich, Carol Tagliaferri, Roy Swonger: 11g Upgrade Paper – Oracle Germany)

- University with about 20,000 users on Sun Solaris
  - Moved 10 databases from 9.2.0.8 to 11.1.0.6
  - Used SQL Tuning Advisor and SQL Performance Analyzer (SPA) to fix 94 queries
  - Also moved to RAC, ASM & Data Guard
  - **30% more logins and yet Response Time is 50% LOWER!**
- International Customer with 400+ databases on IBM AIX & EMC DMX disks
  - Moved from 9.2.0.8 to 11.1.0.6 – 54% slower
  - Used SPM, SPA, DB Replay to tune things…
  - Changed parameters to 11g – 15% improvement
  - Gathered system stats – 7% improvement
  - Used SPA – 18% improvement
  - Turn on SQL Profiling (SPM) – 8% improvement
  - **11g is now 11% FASTER than 9.2.0.8**
- Data Warehouse customer on RH Linux 64-Bit
  - Moving from 10.2 to 11.1.0.7 with 50 databases each at around 10T
  - Over 200,000 partitions in the database
  - Silent Upgrade of 50 other DWHS's unattended using DBUA silent mode

The Future: 8 Exabytes
Look what fits in one 10g Database!

2K – A typewritten page
5M – The complete works of Shakespeare
10M – One minute of high fidelity sound
2T – Information generated on YouTube in one day
10T – 530,000,000 miles of bookshelves at the Library of Congress
20P – All hard-disk drives in 1995 (or your database in 2010)
700P –Data of 700,000 companies with Revenues less than $200M
1E – Combined Fortune 1000 company databases (average 1P each)
1E –Next 9000 world company databases (average 100T each)
8E – Capacity of ONE Oracle10g Database (CURRENT)
12E to 16E – Info generated before 1999 (memory resident in 64-bit)
16E – Addressable memory with 64-bit (CURRENT)
161E – New information in 2006 (mostly images not stored in DB)
1Z – 1000E (Zettabyte - Grains of sand on beaches -125 Oracle DBs)
100TY - 100T-Yottabytes – Addressable memory 128-bit (FUTURE)

8 Exabytes: Look what fits in one 10g Database!

- All databases of the largest 1,000,000 companies in the world (3E).

or

- All Information generated in the world in 1999 (2E)

or

- All Information generated in the world in 2003 (5E)

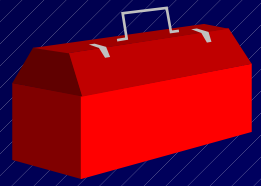
or

- All Email generated in the world in 2006 (6E)

or

- 1 Mount Everest filled with Documents (approx.)

V$ Views over the years



Version	V$ Views	X$ Tables
6	23	? (35)
7	72	126
8.0	132	200
8.1	185	271
9.0	227	352
9.2	259	394
10.1.0.2	340 (+31%)	543 (+38%)
10.2.0.1	396	613

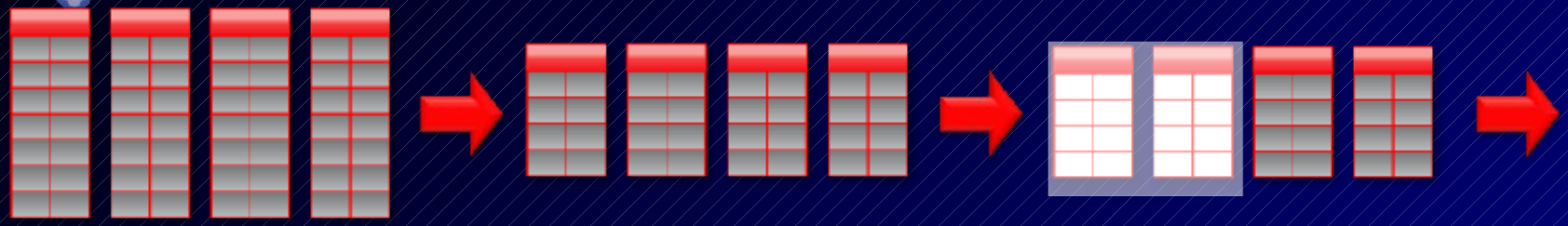
Impact Tuning with Oracle

99.8% Less Data Accessed / 96.8% Time Reduction

Option	Before	After
Partitions	120 sec - 310M	0.43 sec - 200k
Partitions / Tuned	120 sec - 310M	0.01 sec - 8k
Parallel Query (20 Proc.)	230 sec	18 sec
Function-Based Index	1206 sec - 3G	7 sec - 8k
Materialized View	28 sec	3 sec
Cursor_Sharing	240 sec	0.01 sec
Truncate	510 sec / 8G	0.40 sec / 32k
Driving Table	900 sec	1 second
SGA Sizing	30 sec	0.01 sec
750,000 Query Mix	5.1 T / 540 hours	9 G / 23 hours

NOW Oracle will do all of this for you!!!

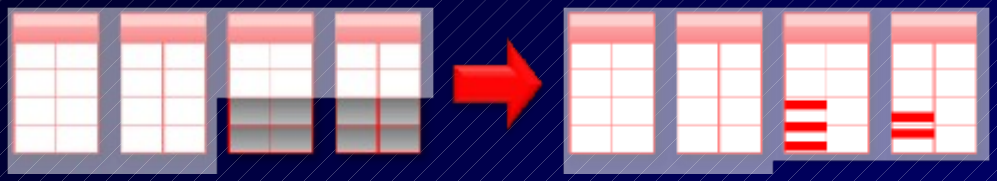
Benefits Multiply*



10 TB of user data
Requires 10 TB of IO

1 TB
with compression

100 GB
with partition pruning



20 GB
with Storage Indexes

5 GB
with Smart Scans

Sub second
On Database Machine

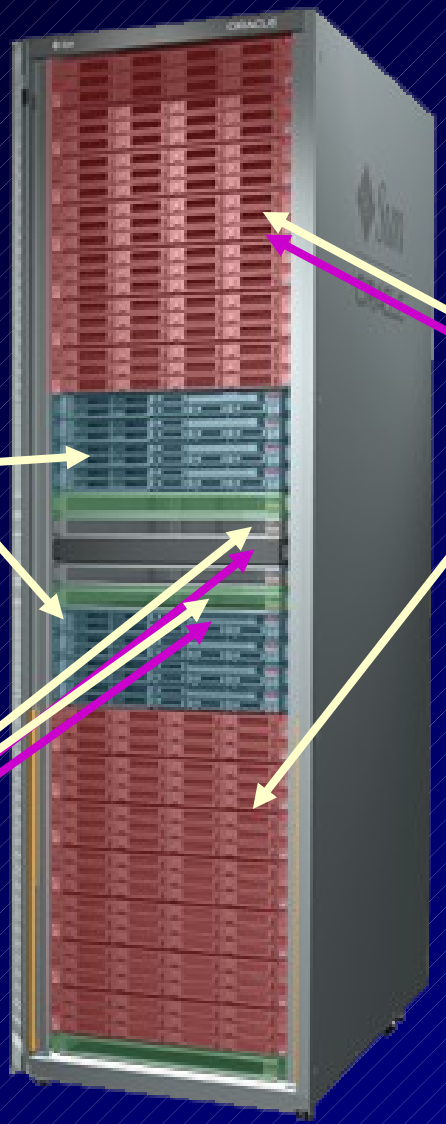
Data is 10x Smaller, Scans are 2000x faster

*Oracle Slide –

Easier way – Oracle's picture of Exadata V2 (X2-2)!

8 Compute Servers

- 8 x 2 sockets x 4 cores = 64 cores
- 576 GB DRAM



14 Storage Servers

- 14x12=168 Disks
- 100T SAS or
- 336T SATA

- 5TB+ flash storage!

InfiniBand Network

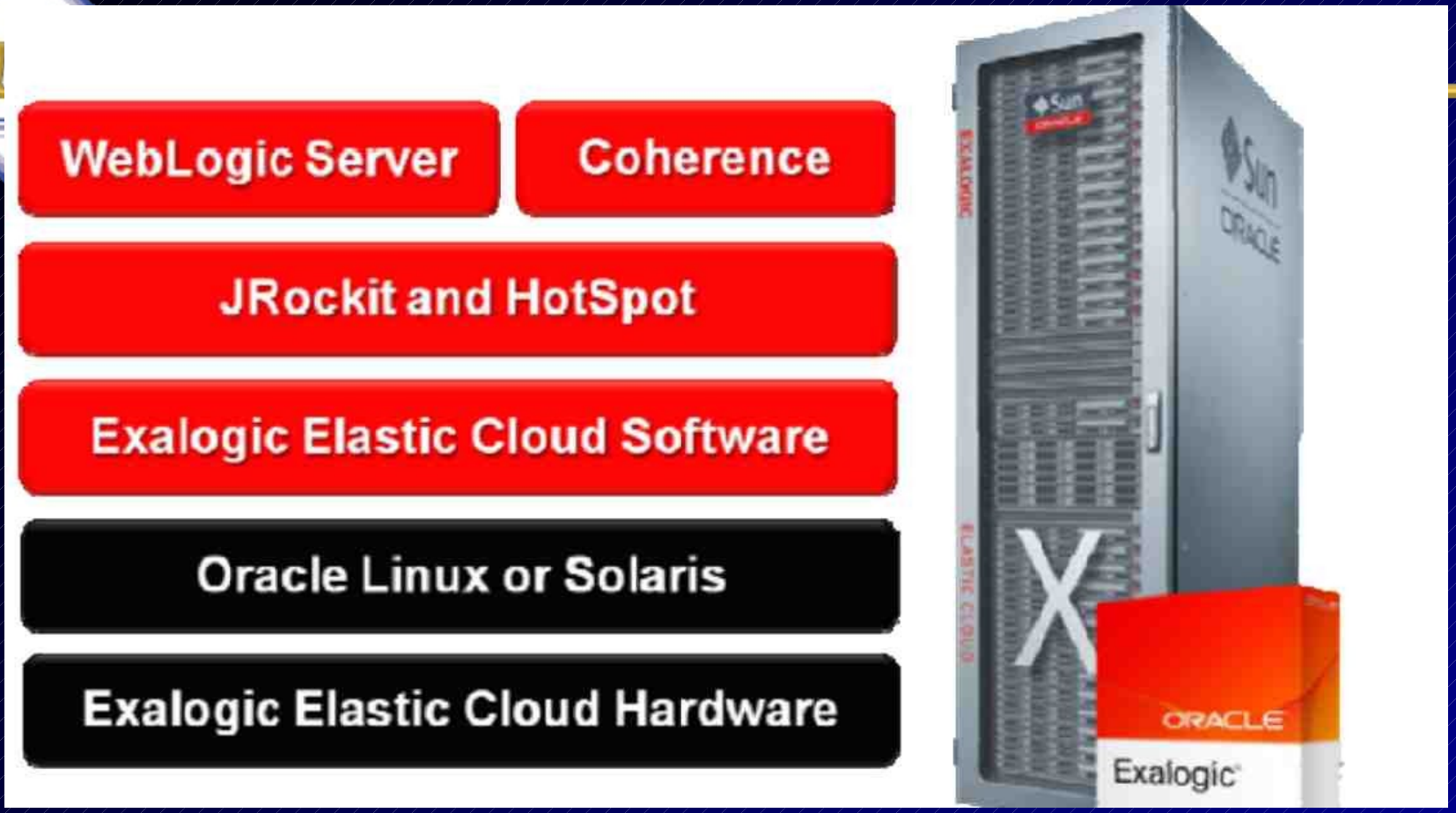
- 40 Gb/sec each direction
- Fault Tolerant

What's Next – Exadata X2-8

- 2 compute servers (7560 CPU at 2.26 GHz & 5T SAS)
  - 2 servers x 8 CPU sockets x 8 cores = **128 cores**
- 2 compute servers x 1T DRAM = **2T DRAM**
- **Same storage numbers…**

(FUTURE?? 8 servers = 512 CPUs & 8T of DRAM)

What's Next – Exalogic Elastic Cloud!



- Some points here – Leveraging those acquisitions!
  - Coherence is a great product / NEW Linux – **Unbreakable Enterprise Kernel!**
  - 360 CPUs, 2.8T DRAM, 980G FlashFire SSD, 40T SAS – Will help Fusion Apps Smoke!
  - 1M HTTP/sec – could fit Facebook on 2 of these even thought there are 500M people on Facebook

Exadata = Paradigm Shift!

Oracle is never caught from behind
Oracle's 30th Anniversary in 2007

- Great Sales/Marketing
- Great Database
- Applications Leader
- BI Leader
- Already in the lead
- GAME OVER

A Diverse Team is Oracle's Secret!

"Larry Ellison is the genius behind Oracle, the company, Bob Miner was the genius behind Oracle, the product. The combination of the diverse team Oracle has had over the years is the secret of their success!"

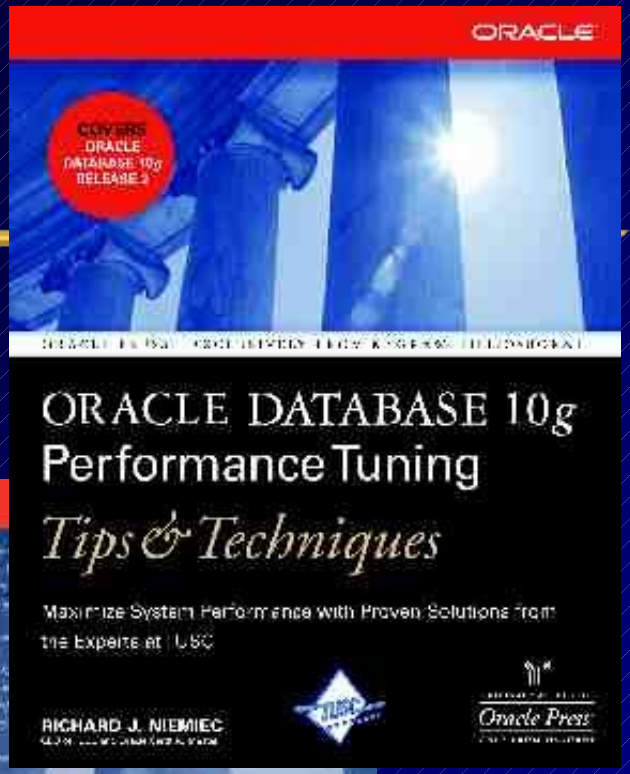
- Rich Niemiec, Select Magazine, 2001

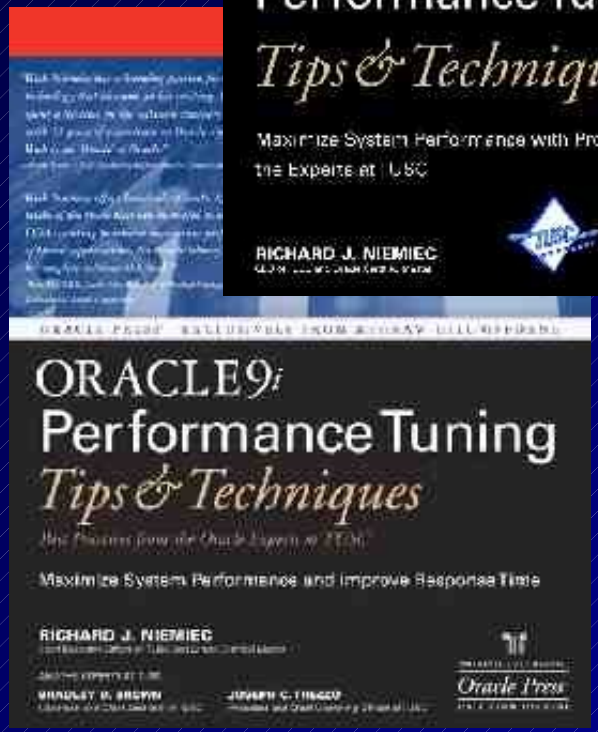
Summary

- Know the Oracle
- Start Me Up – Using Memory Target
- The Buffer Cache & The Result Cache
- Virtual Columns
- Invisible Indexes & Online Index Rebuilds
- Creating & Rebuilding Indexes Online
- Secure Files
- DDL Lock Timeout, PL/SQL Expressions/Simple Integer
- ADDM Enhancements
- SQL Plan Management (SPM) and capturing SQL Plan Baselines
- SQL Performance Analyzer, Access Advisor & Query Repair Advisor
- Real Application Testing (Database Capture and Replay)
- Interval Partitioning & Partition Compression
- Automatic Diagnostic Repository (ADR)
- Auto Sample, Creating Pending Statistics
- Adaptive Cursor Sharing and Bind Peeking
- EM, Grid Control, Security Enhancements & the Future Sizes

For More Information

- **www.tusc.com**
- *Oracle9i Performance Tuning Tips & Techniques; Richard J. Niemiec; Oracle Press (May 2003)*
- *Oracle 10g Tuning (June 11, 2007)*

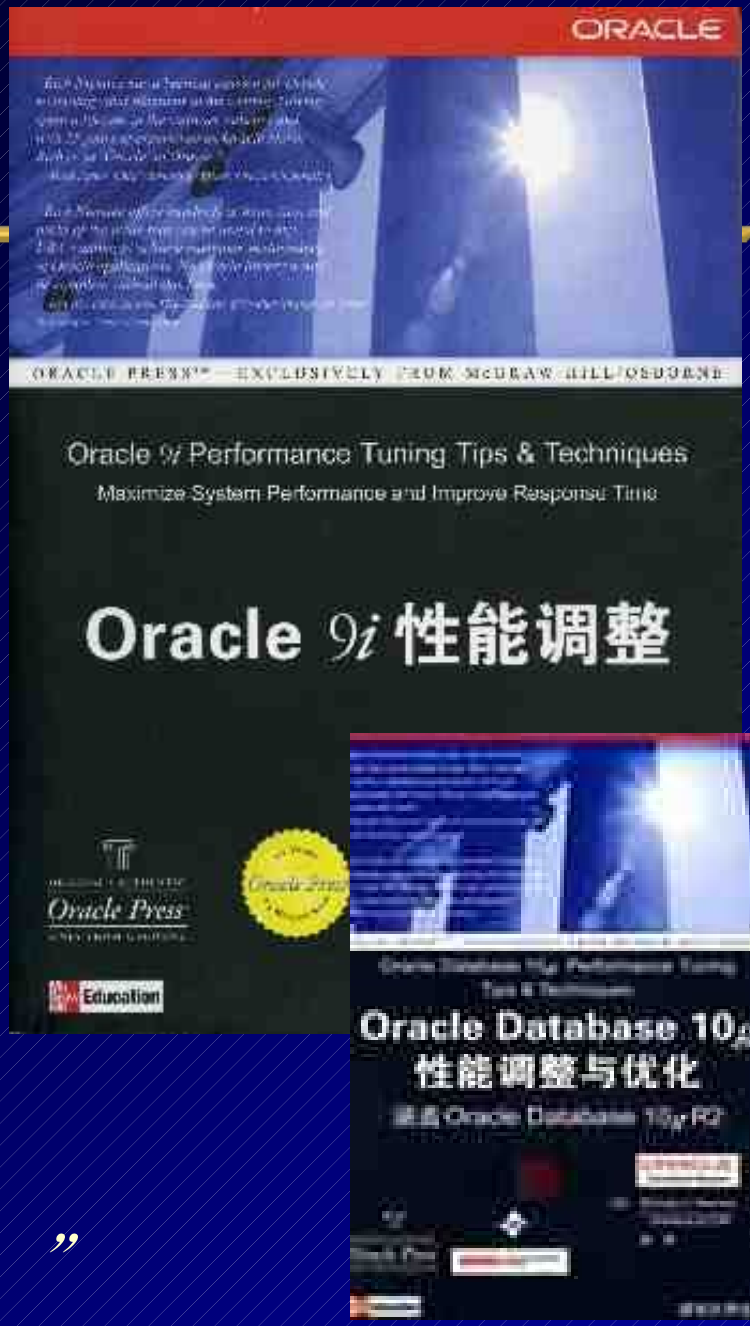




"If you are going through hell, keep going" - Churchill

更多信息

- www.tusc.com
- *Oracle9i Performance Tuning Tips & Techniques; Richard J. Niemiec; Oracle Press (May 2003)*
- *Oracle 10g Tuning (June 11, 2007)*



“成功只访问那些没空追求它的人。”

- Henry David Thoreau

"You must BE the change you want to see in the world."

--Mahatma Gandhi

References

- www.tusc.com. www.rolta.com
- *Oracle10g Performance Tuning Tips & Techniques; Richard J. Niemiec; Oracle Press*
- Database Secure Configuration Initiative: Enhancements with Oracle Database 11g, www.oracle.com
- All Oracle11g Documentation from Oracle Beta Site
- Introduction to Oracle Database 11g, Ken Jacobs
- Oracle Database 11g New Features, Linda Smith
- New Optimizer Features in 11g, Maria Colgan
- www.ioug.org, www.oracle.com, en.wikipedia.org & technet.oracle.com

- Thanks Dan M., Bob T., Brad, Joe, Heidi, Mike K., Debbie, Maria, Linda

- **Dedicated to the memory of Robert Delgado Patton, Stan Yellott, Mark Beaton, Ray Mansfield, Lex De Haan, Elaine DeMeo and Jim Gray**

Rolta TUSC – *Your* Partner …. Accomplished in Oracle!

2010 Oracle Partner of the Year (7 Titans Total)

Prior Years Winner 2002, 2004*, 2007*, 2008

*Won 2 Awards

Rolta TUSC Services

- **Oracle**
  - E-Business Suite implementation, R12 upgrades, migration & support
  - Fusion Middleware and Open Systems development
  - Business Intelligence (OBIEE) development
  - Hyperion Financial Performance Management
  - DBA and Database tactical services
  - Strategic Global Sourcing
- **IT Infrastructure**
  - IT Roadmap - Security & Compliance - Infrastructure Management
  - Enterprise Integration / SOA - High Availability and Disaster Planning
- **Profitability & Cost Management**
  - Financial Consolidation - Budgeting & Forecasting
  - Profitability & Risk Analysis - Enterprise Performance Management
  - Operational, Financial & Management Reporting
- **Rolta Software Solutions**
  - *i*Perspective™ - rapid data & systems integration
  - Geospatial Fusion™ - spatial integration & visualization
  - OneView™ - business & operational intelligence

Copyright Information

- Neither Rolta TUSC nor the author guarantee this document to be error-free. Please provide comments/questions to rich@tusc.com. I am always looking to improve!
-

Contact Information

Rich Niemiec: rich@tusc.com

www.tusc.com

Dedicated to the Memory of Robert Delgado Patton

Rich's Overview (rich@tusc.com)

- Advisor to Rolta International Board
- Former President of TUSC
  - Inc. 500 Company (Fastest Growing 500 Private Companies)
  - 10 Offices in the United States (U.S.); Based in Chicago
  - Oracle Advantage Partner in Tech & Applications
- Former President Rolta TUSC & President Rolta EICT International
- Author (3 Oracle Best Sellers – #1 Oracle Tuning Book for a Decade):
  - Oracle Performing Tips & Techniques (Covers Oracle7 & 8i)
  - Oracle9i Performance Tips & Techniques
  - Oracle Database 10g Performance Tips & Techniques
- Former President of the International Oracle Users Group
- Current President of the Midwest Oracle Users Group
- Chicago Entrepreneur Hall of Fame - 1998
- E&Y Entrepreneur of the Year & National Hall of Fame - 2001
- IOUG Top Speaker in 1991, 1994, 1997, 2001, 2006, 2007
- MOUG Top Speaker Twelve Times
- National Trio Achiever award - 2006
- Oracle Certified Master & Oracle Ace Director
- Purdue Outstanding Electrical & Computer and Engineer - 2007