# The Best Oracle Database 11g New Features

## New York 2009

*Rich Niemiec, Rolta TUSC (www.tusc.com)*

*(Thanks: Caryl Lee Fisher, Paul Dorsey, Ken Jacobs, Debbie Migliore, Maria Colgan, Linda Smith)*

# Audience Knowledge / Versions

- Oracle7 Experience ?
- Oracle8*i* Experience ?
- Oracle9*i* Experience ?
- Oracle10g Experience?
- Oracle Database 11g Experience?


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

# Overview

- Start Me Up – Using Memory Target
- The Result Cache
- Invisible Indexes & Online Index Rebuilds
- Nice Developer Features
- ADDM Enhancements
- SQL Plan Management and SQL Plan Baselines
- SQL Query Repair Advisor
- SQL Performance Analyzer
- Real Application Testing (Database Capture and Replay)
- Interval Partitioning & Partition Compression
- DBA Tools and DBMS_STATS Enhancements
- Grid Control & EM
- Security Enhancements & the Future Sizes
- Summary

# 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!**

# 2007: Version 11g Arrive



- 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
- New releases of Siebel, PeopleSoft and Oracle12 Apps.
- New Oracle BI Suite & **Acquisition of Hyperion**

# Testing the Future Version

## Version 11.1.0.6.0 of the Database

# 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!

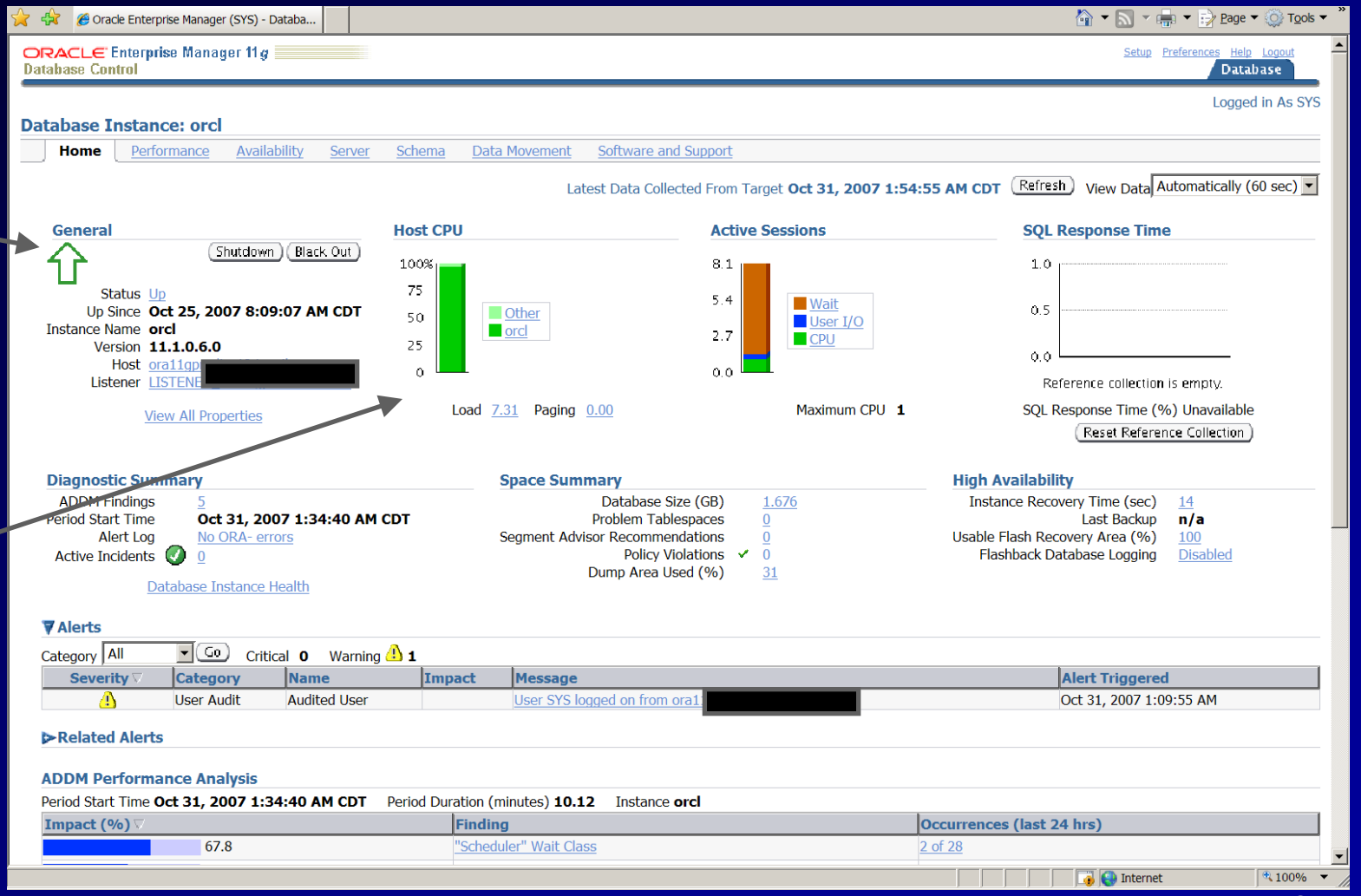


```
$ sqlplus ***/***

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.
```

# Database Information - UP!

Monitor Database (UP)

Users are Definitely Using it!



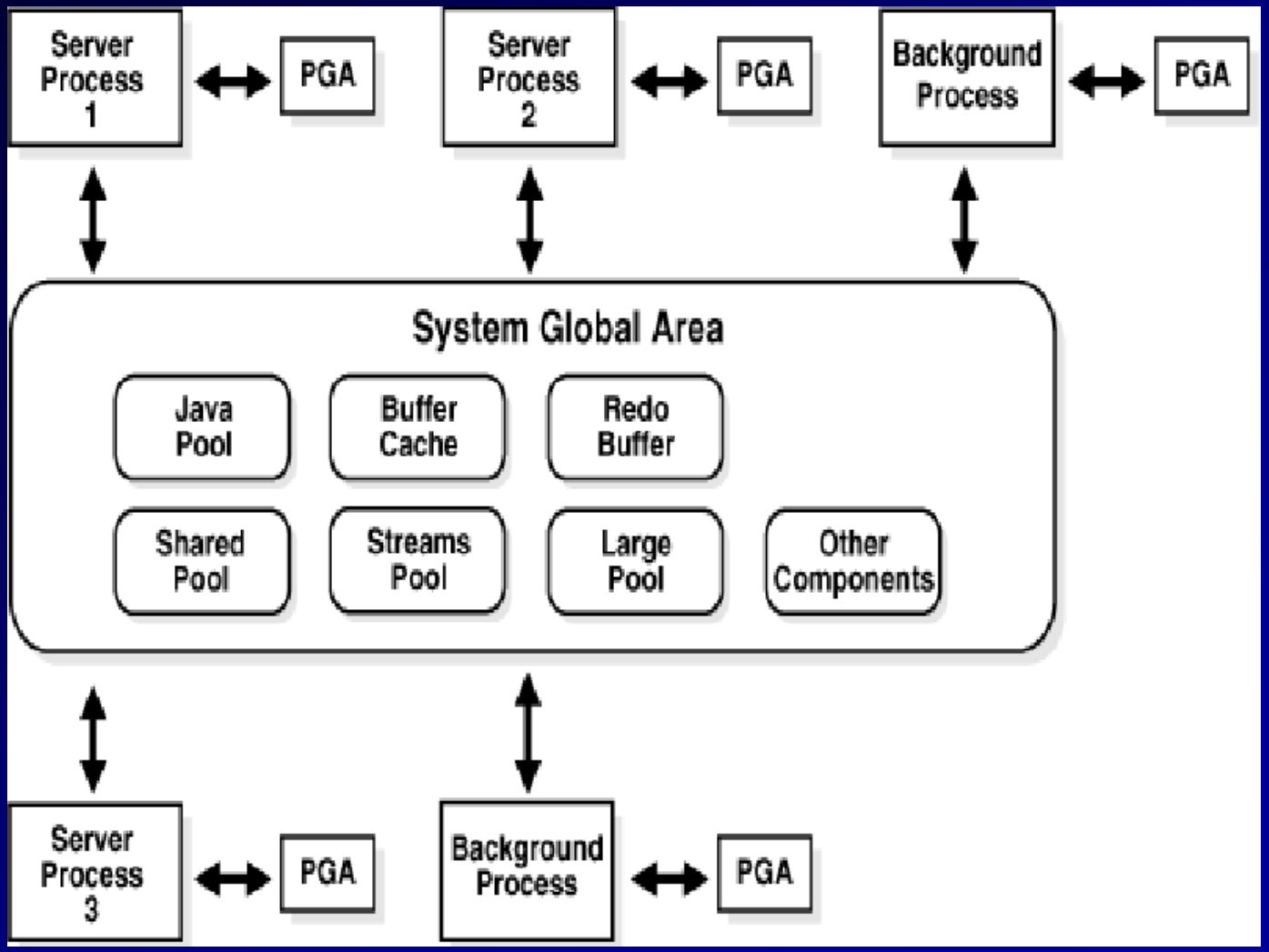
# MEMORY_TARGET
# &
# Automatic Memory Management

# Automatic Memory Management (AMM)
## MEMORY_TARGET in 11g

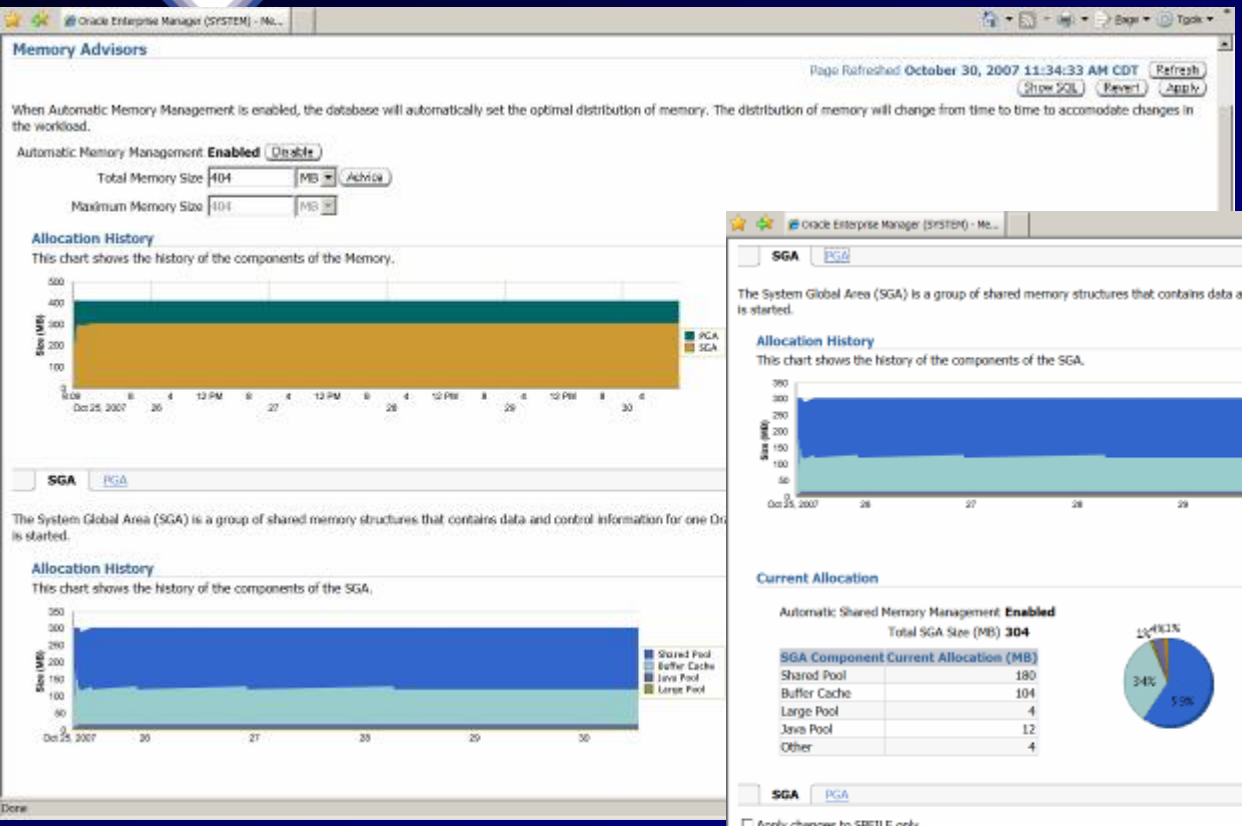


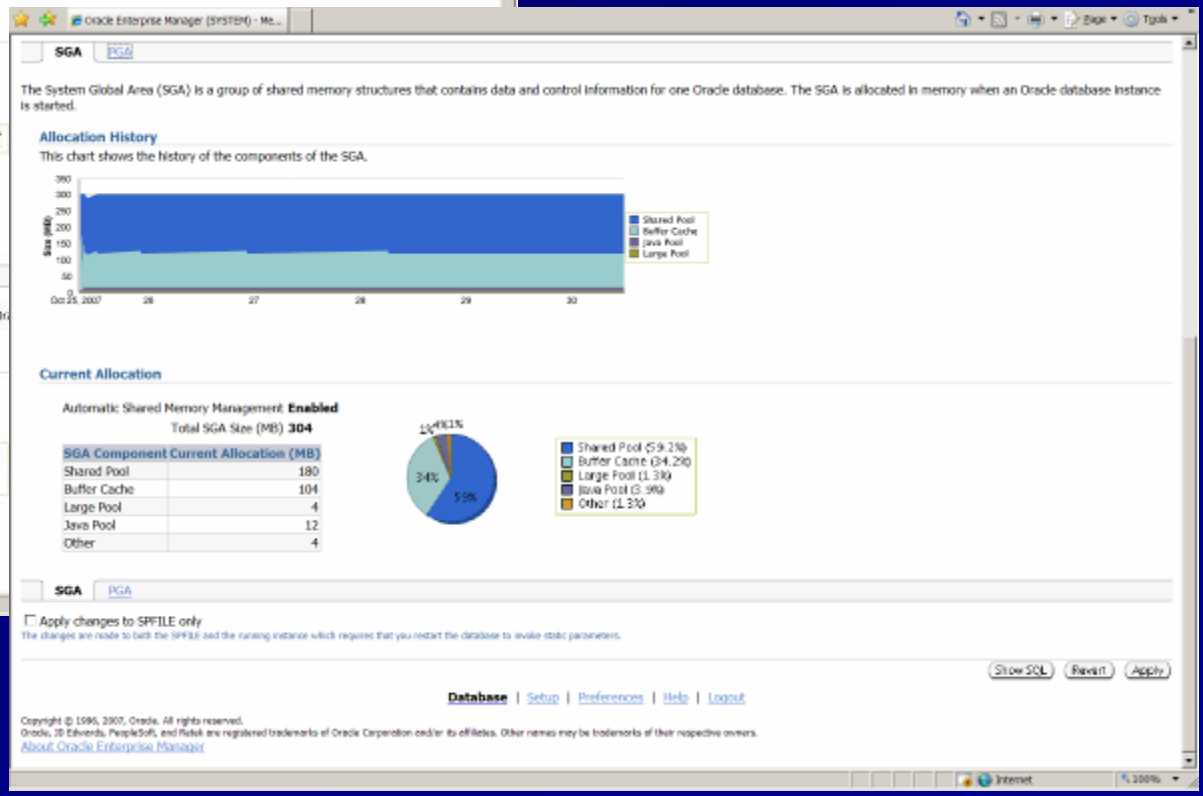
- First there was some Automatic Memory Mgmt - 9i
  - SGA_MAX_SIZE introduced in 9i – Dynamic Memory
  - No more Buffers – DB_CACHE_SIZE
  - Granule sizes introduced - _ksm_granule_size
- Then came SGA_TARGET – 10g
  - Oracle Applications recommends setting this for SGA
  - Set minimums for key values (Data Cache / Shared

# SGA & PGA will be MEMORY_TARGET



# Moving from SGA_TARGET to: MEMORY_TARGET - EM



# ADDM Enhancements
## (Automatic Database Diagnostic Monitor)

# ADDM enhancements

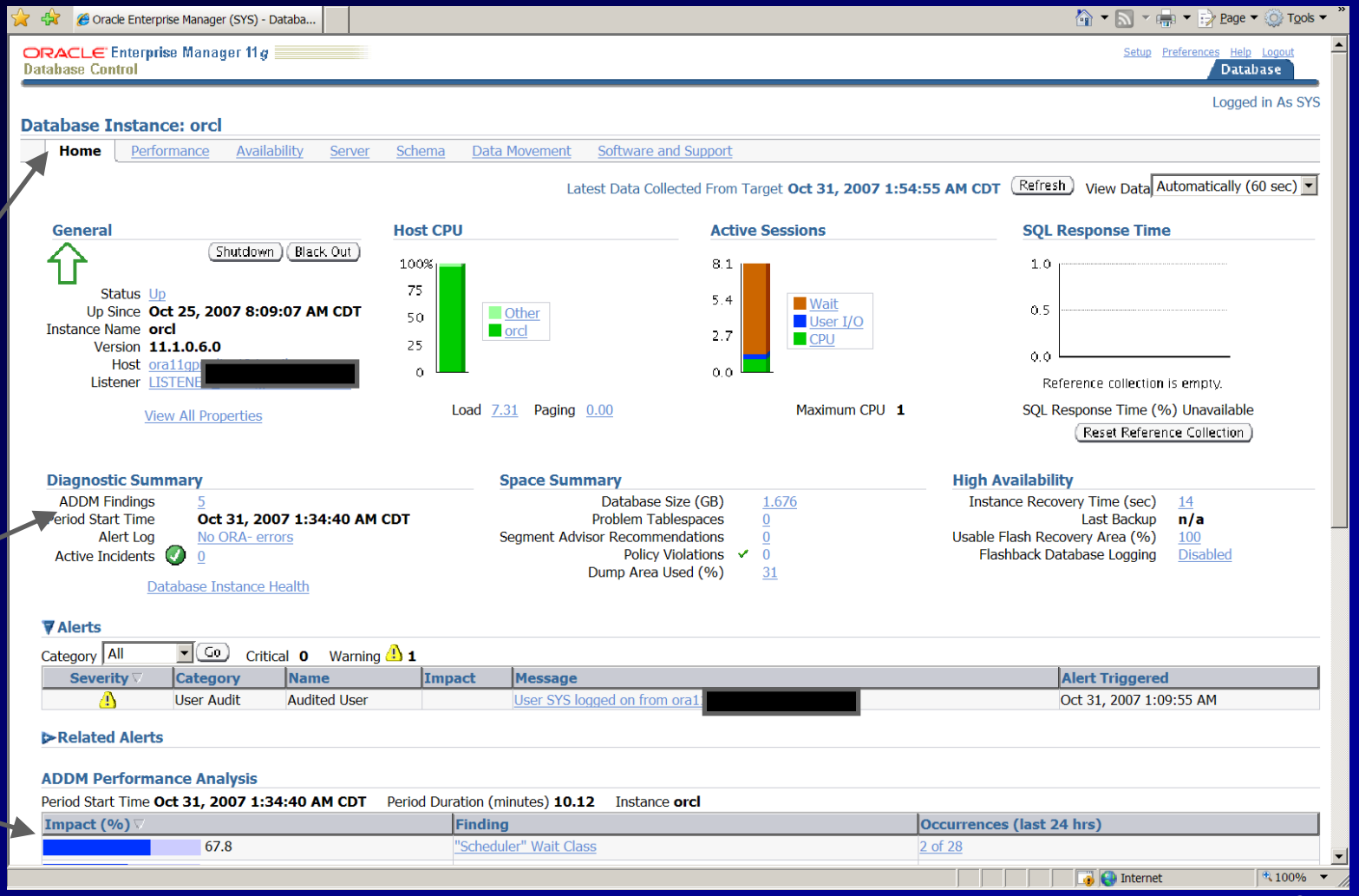
- Global ADDM so that Diagnostics are done across the entire cluster
- Emergency ADDM for use when database is hung
- On any granularity
  - Database Cluster
  - Database Instance
  - Specific Target (such as host, ASM...etc.)
- Over a specified time NOT tied to a pair of

# ADDM Briefly

Specific Database Instance

We have 5 ADDM Findings

Check them Here

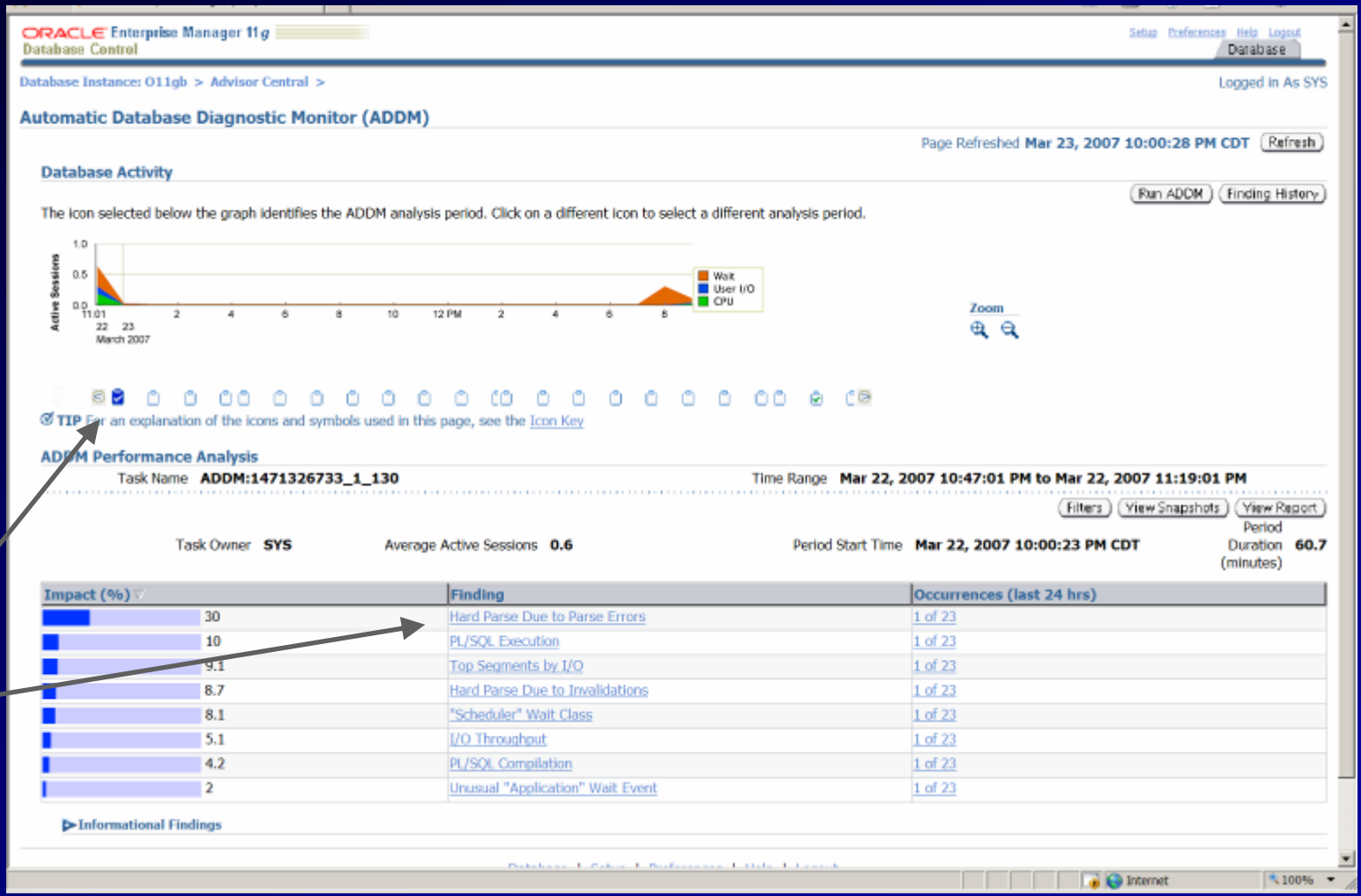


# ADDM Briefly

Top ADDM Findings

Click a Single Timeframe

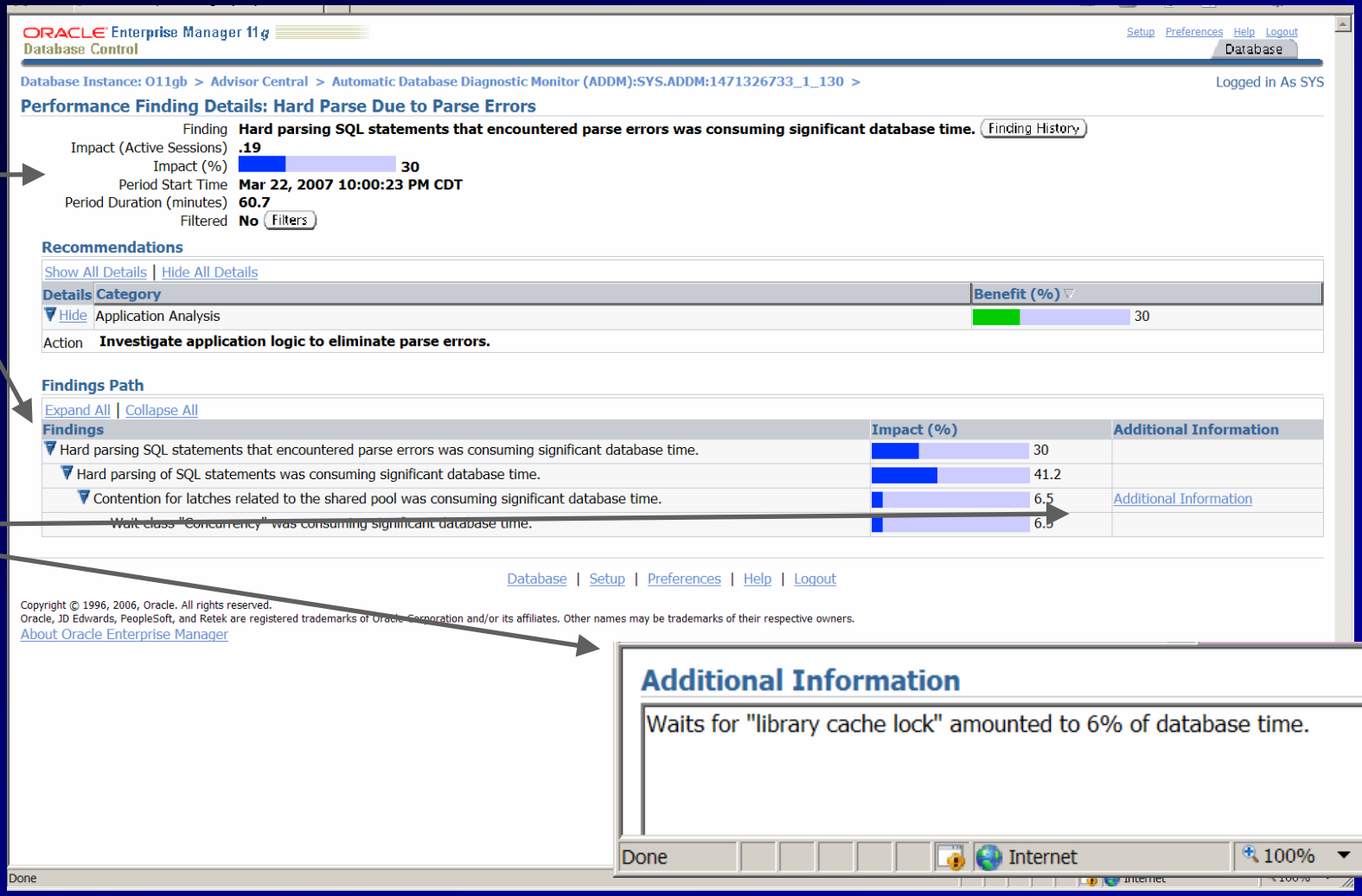
Let's Check the Hard Parse Issue



# ADDM Briefly

Detailed Info & Findings

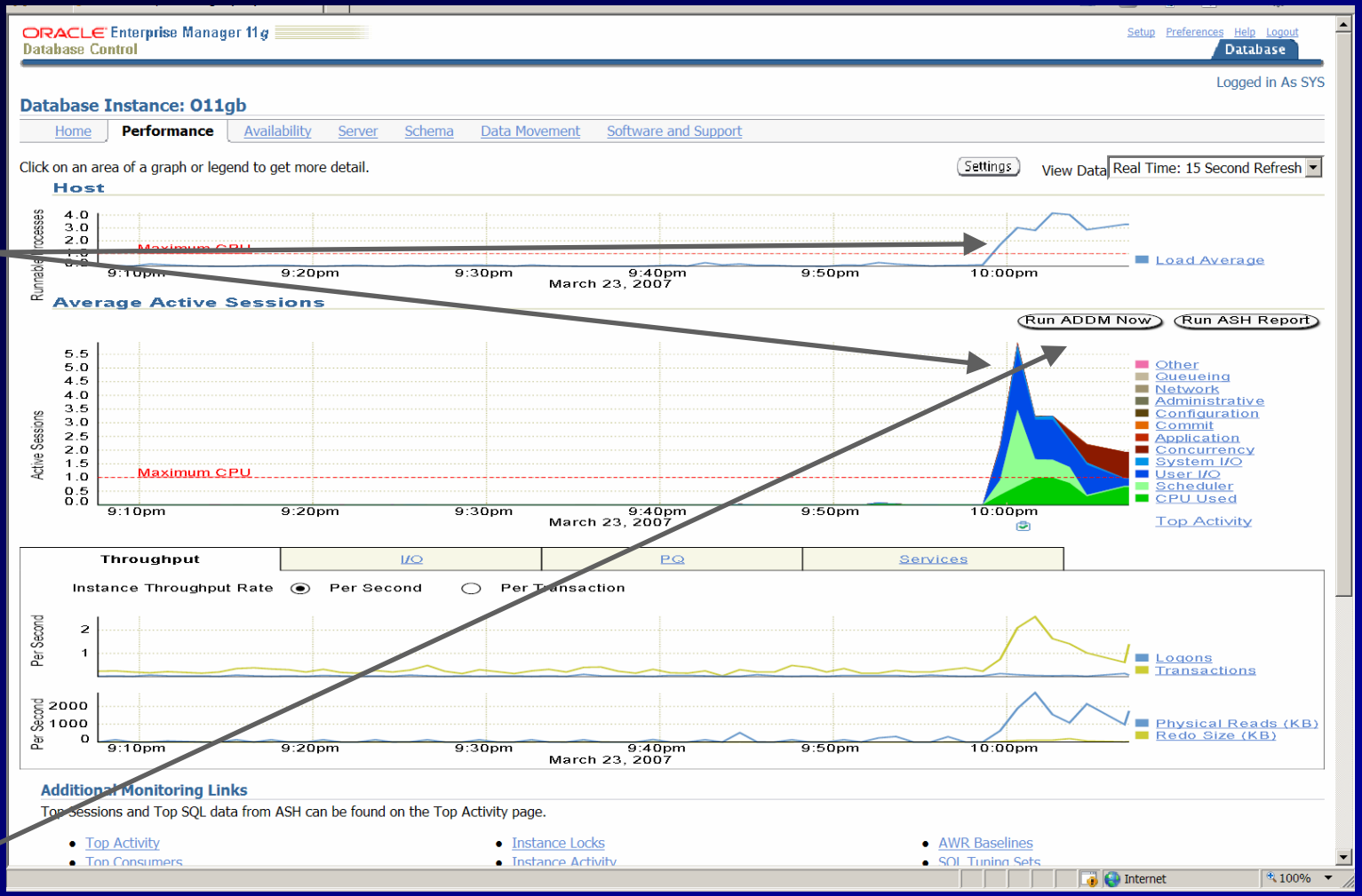
Add'l Info



# ADDM - Run NOW!

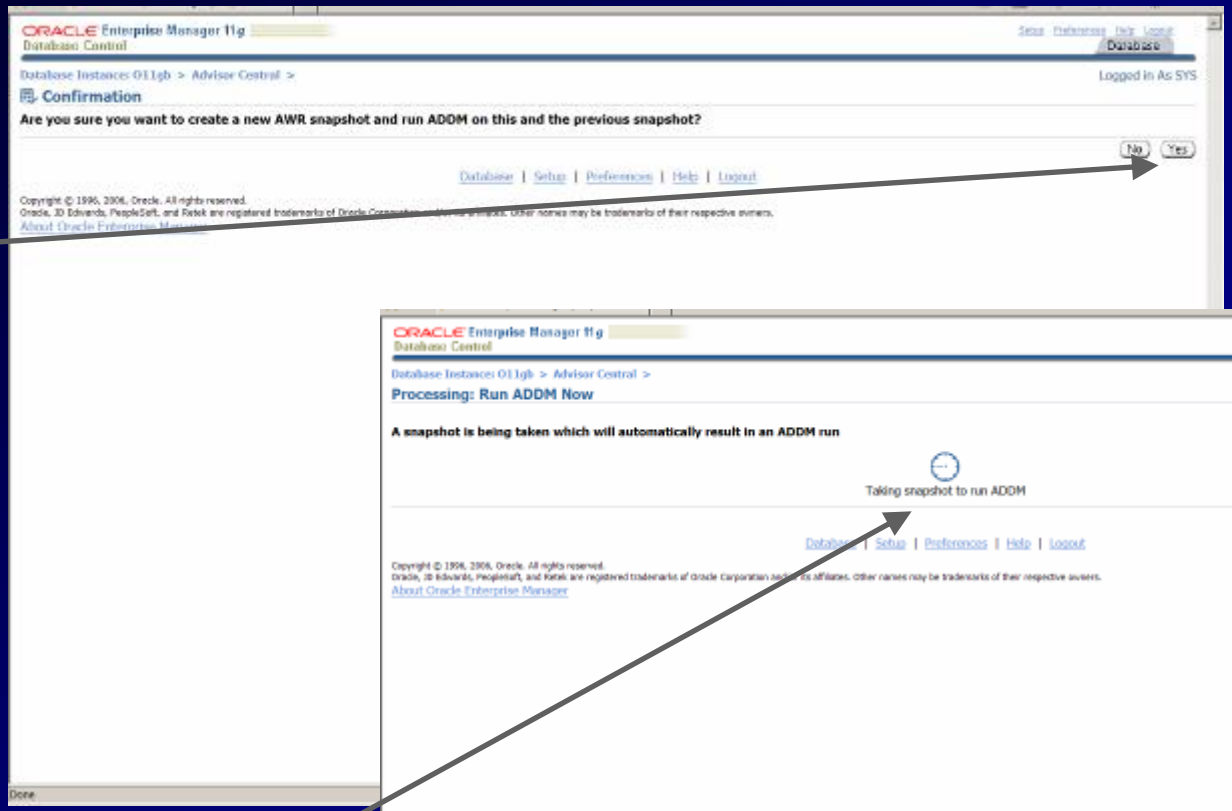
A Big Problem Occurs

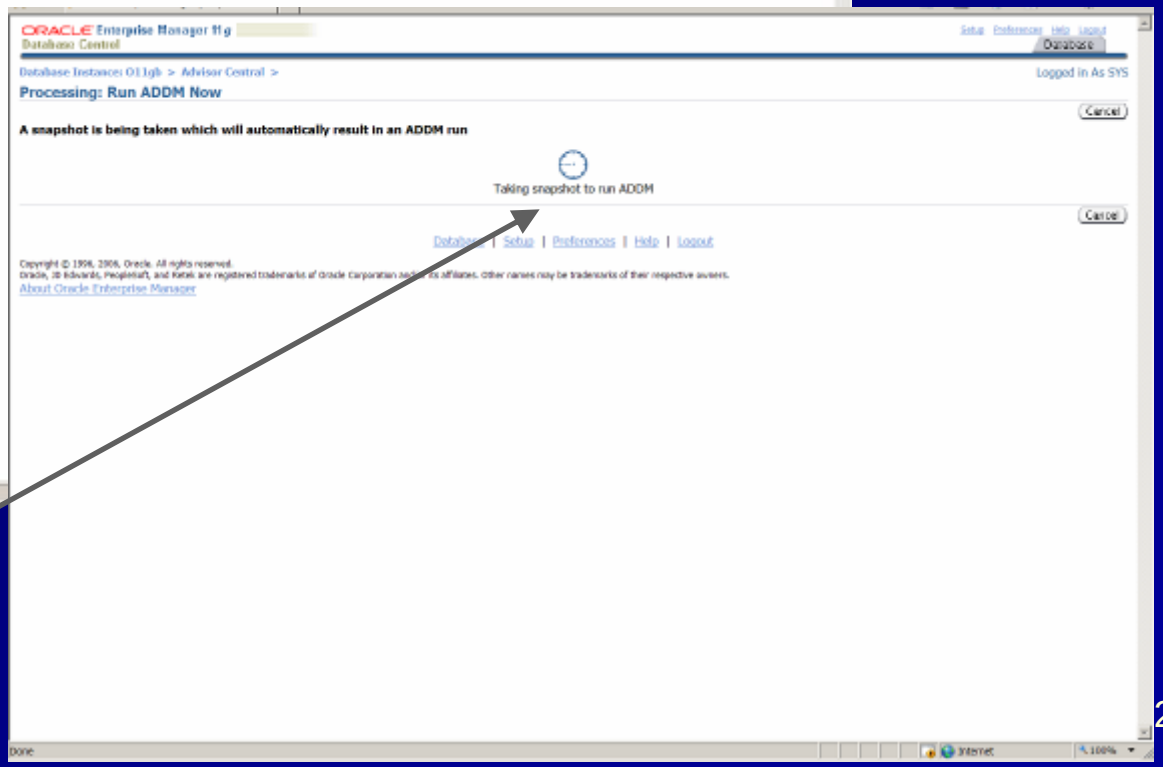
Run ADDM NOW!



# ADDM - Run NOW!

Are you Sure?



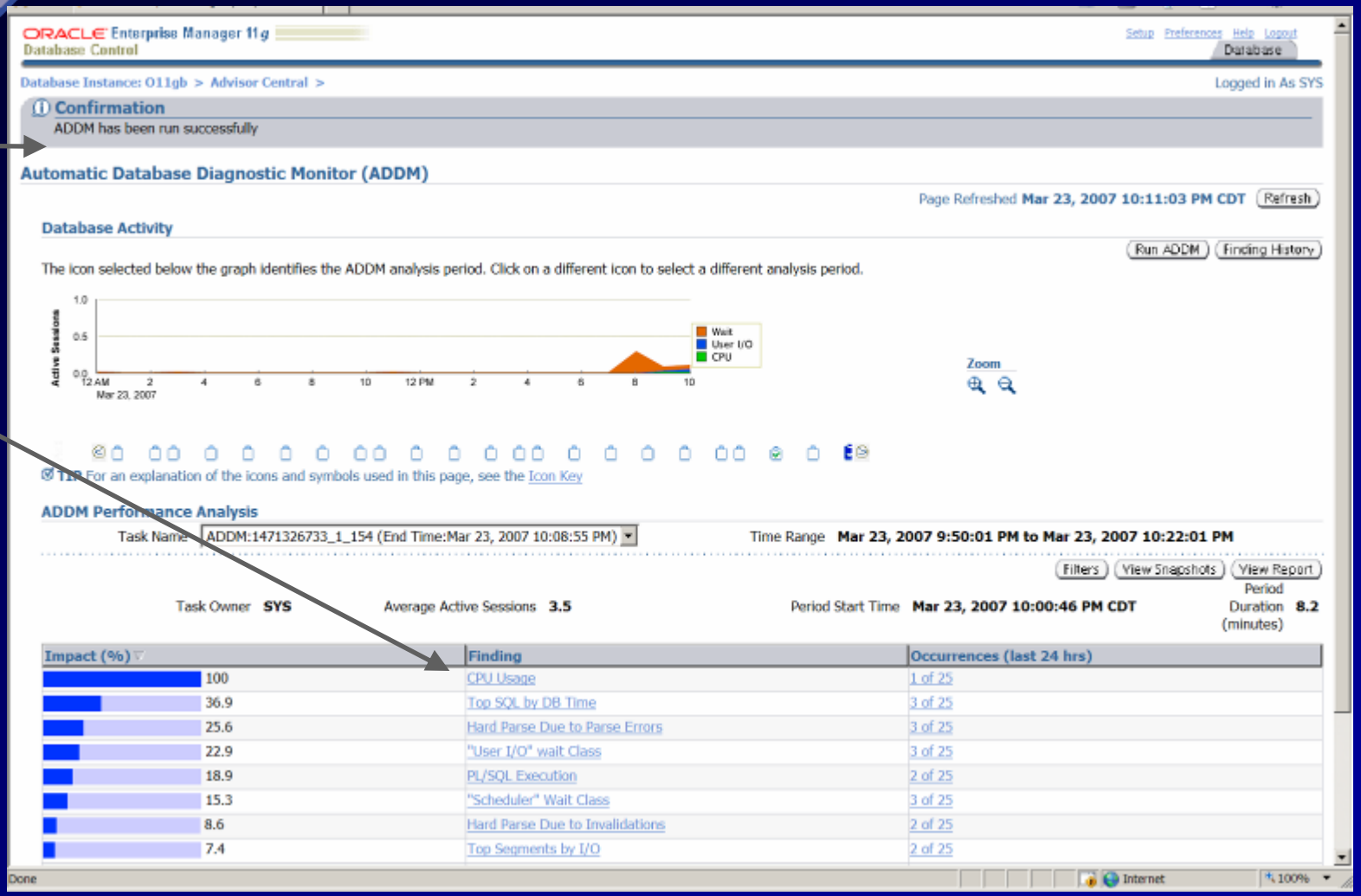




Running

# ADDM – Run NOW!

Done.

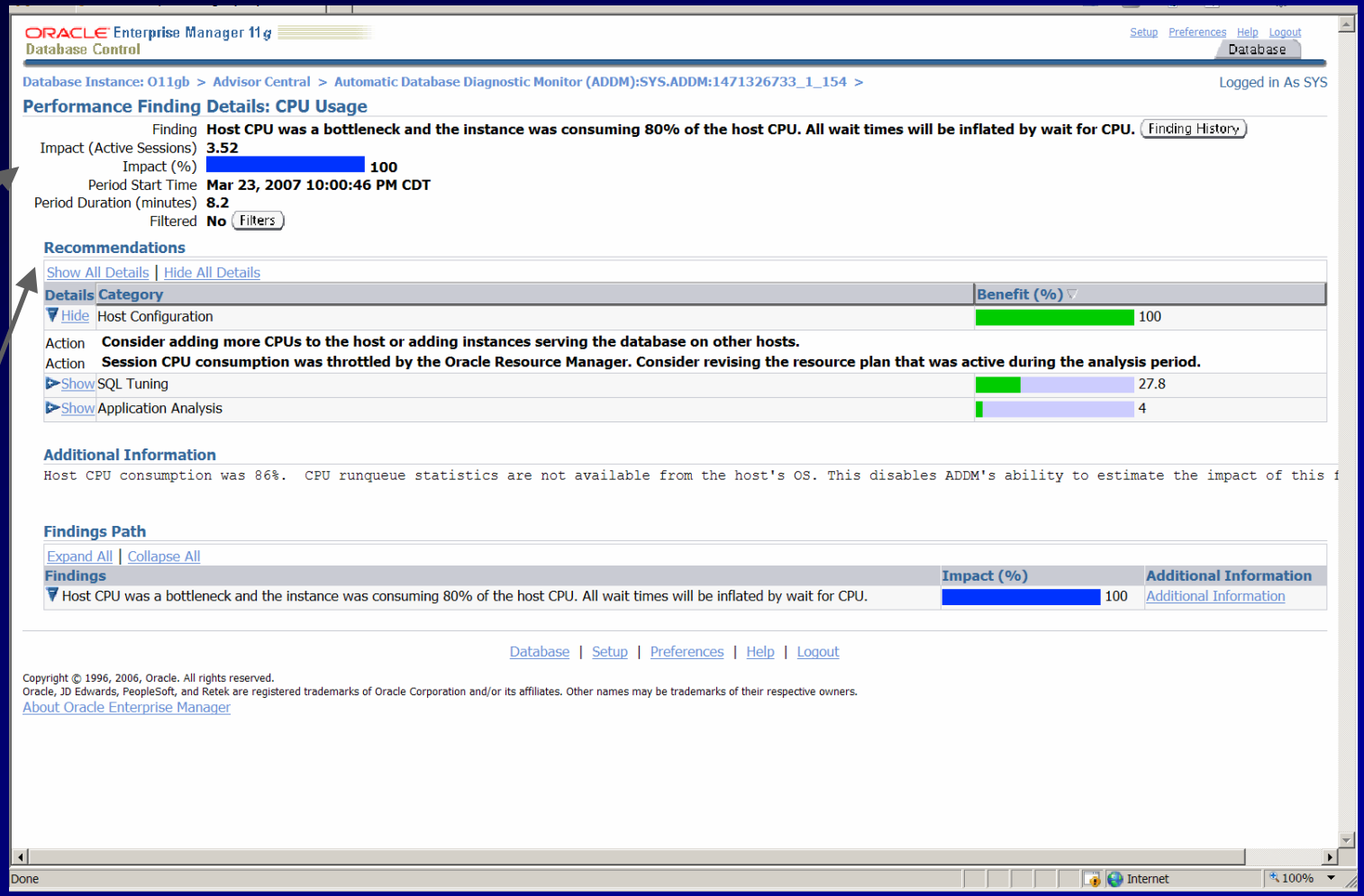
CPU Issue



# ADDM – Run NOW!

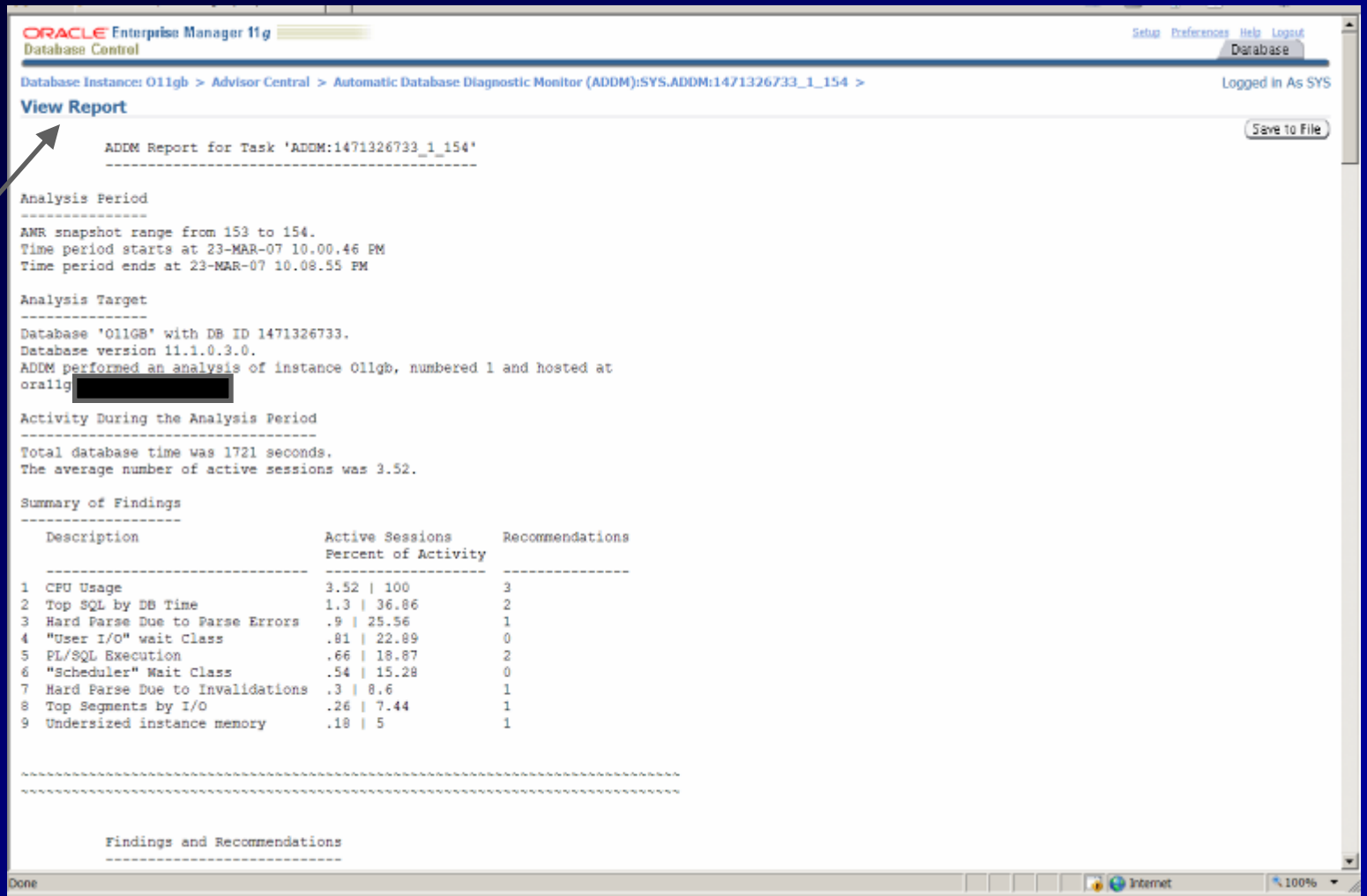
Detail on CPU Issue?

Suggested Fixes

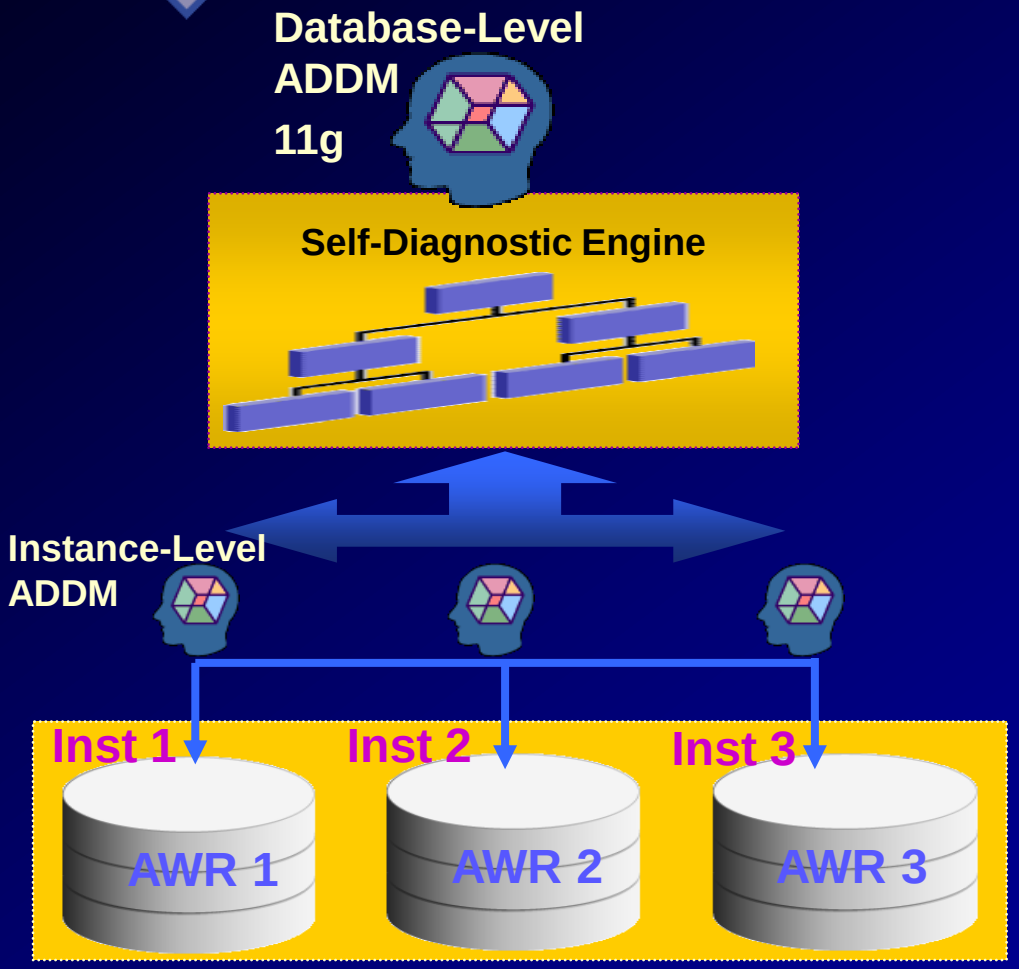


# ADDM – Run NOW!

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

# 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 11g. 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

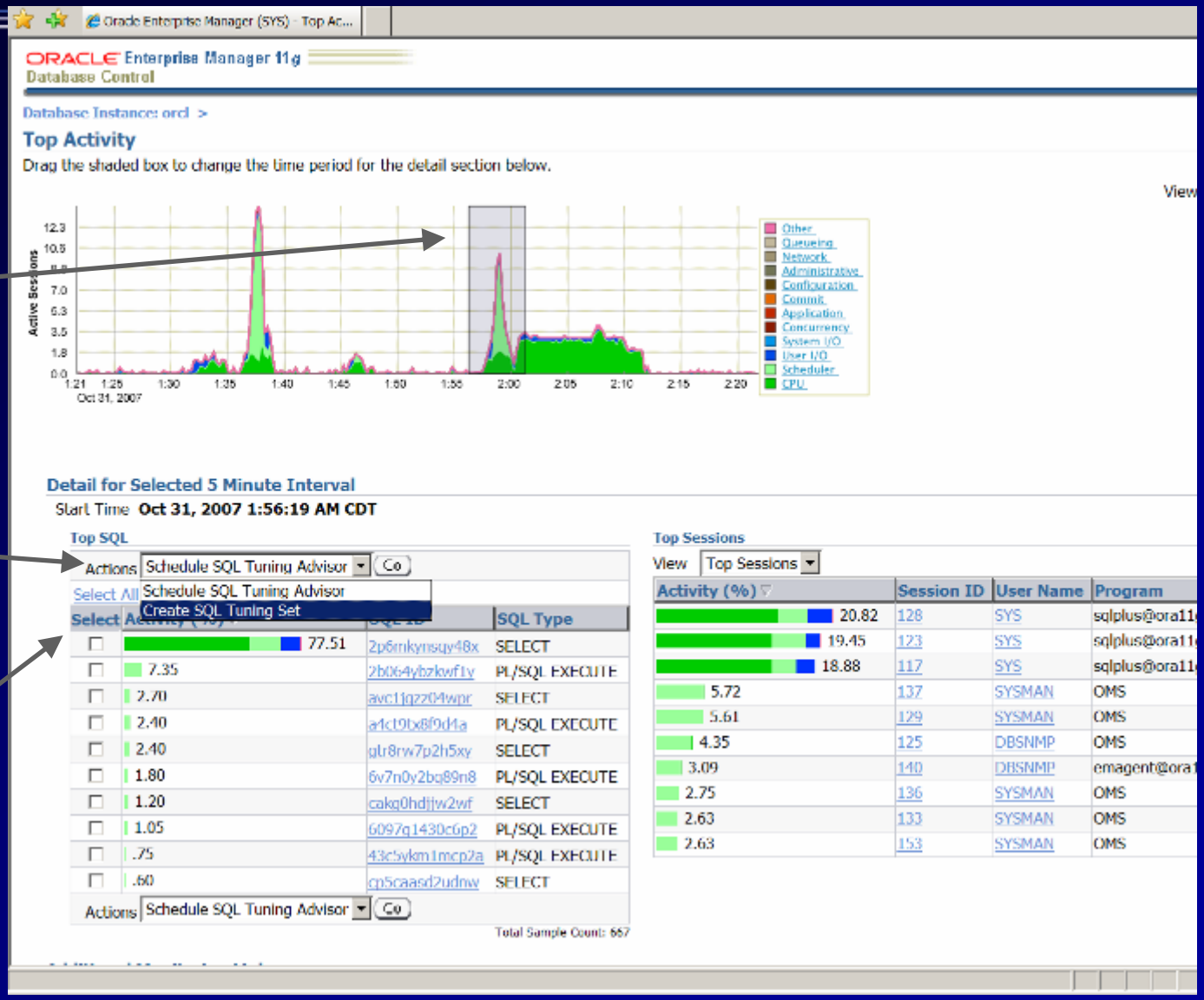
# SQL Plan Management

- **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

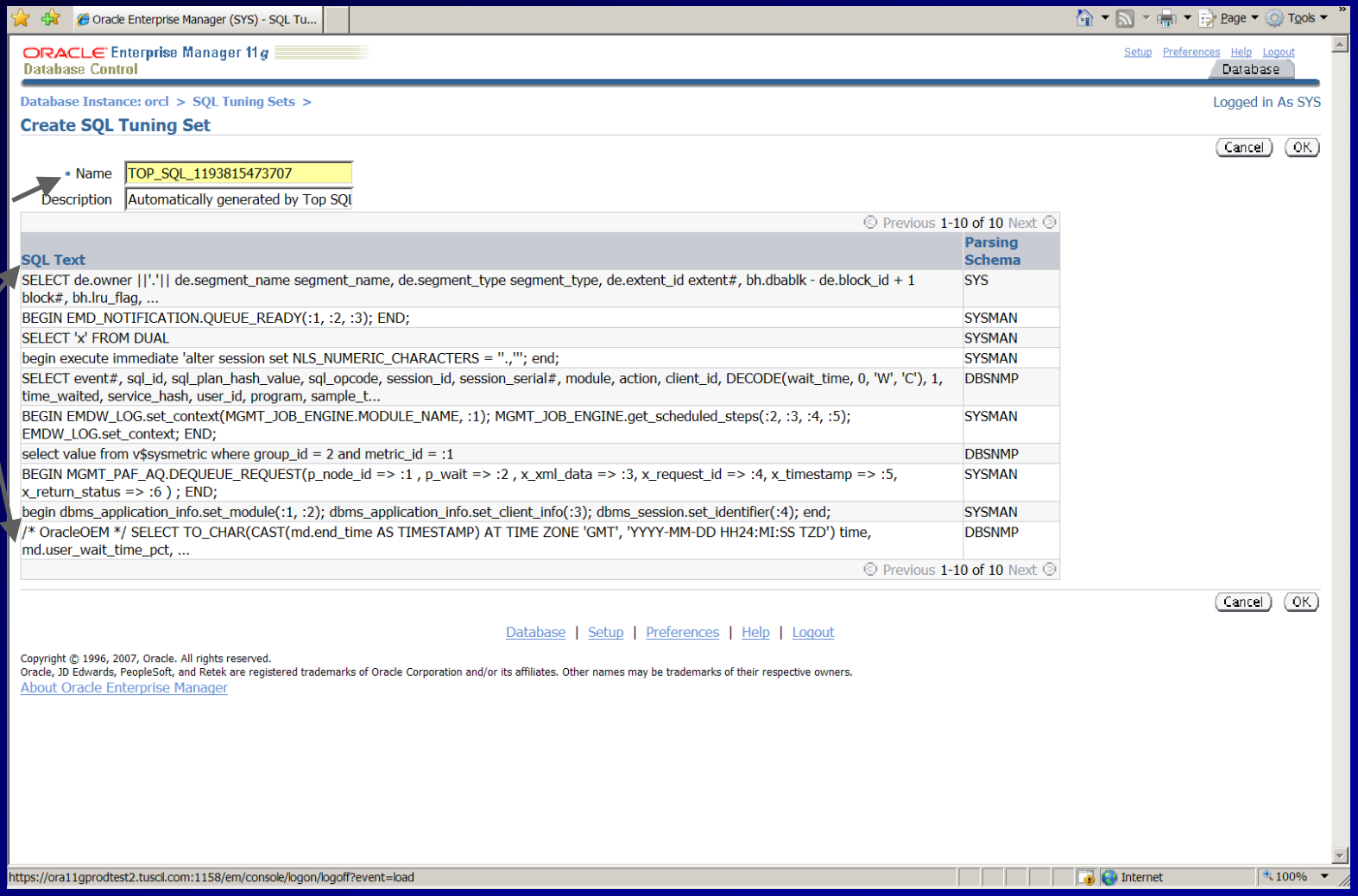
Create a Tuning Set from Top 10 SQL



# SQL Plan Management
# Create a SQL Tuning Set

Tuning Set Name

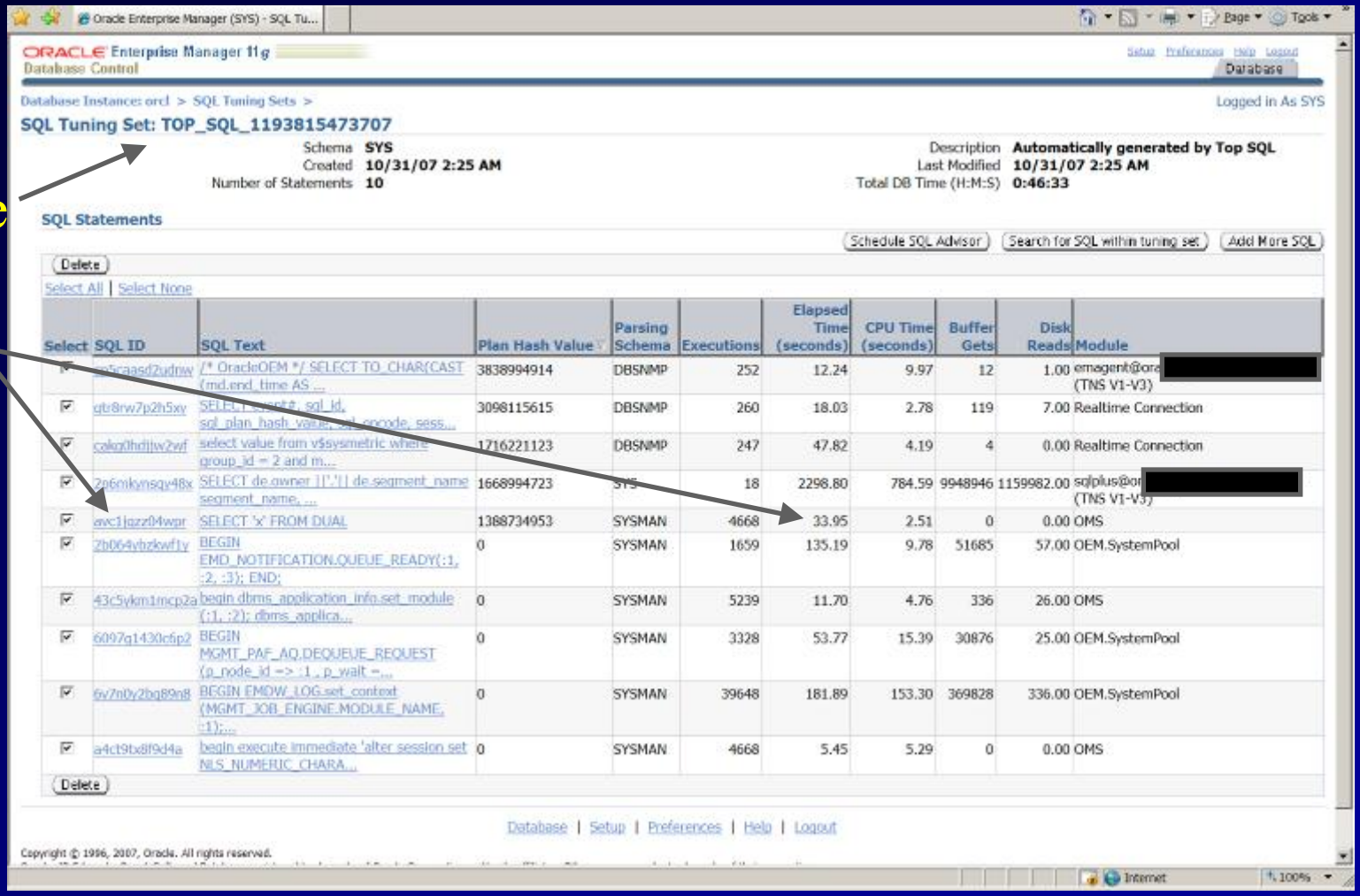
Queries



# SQL Plan Management
# Viewing a SQL Tuning Set

Tuning Set Name

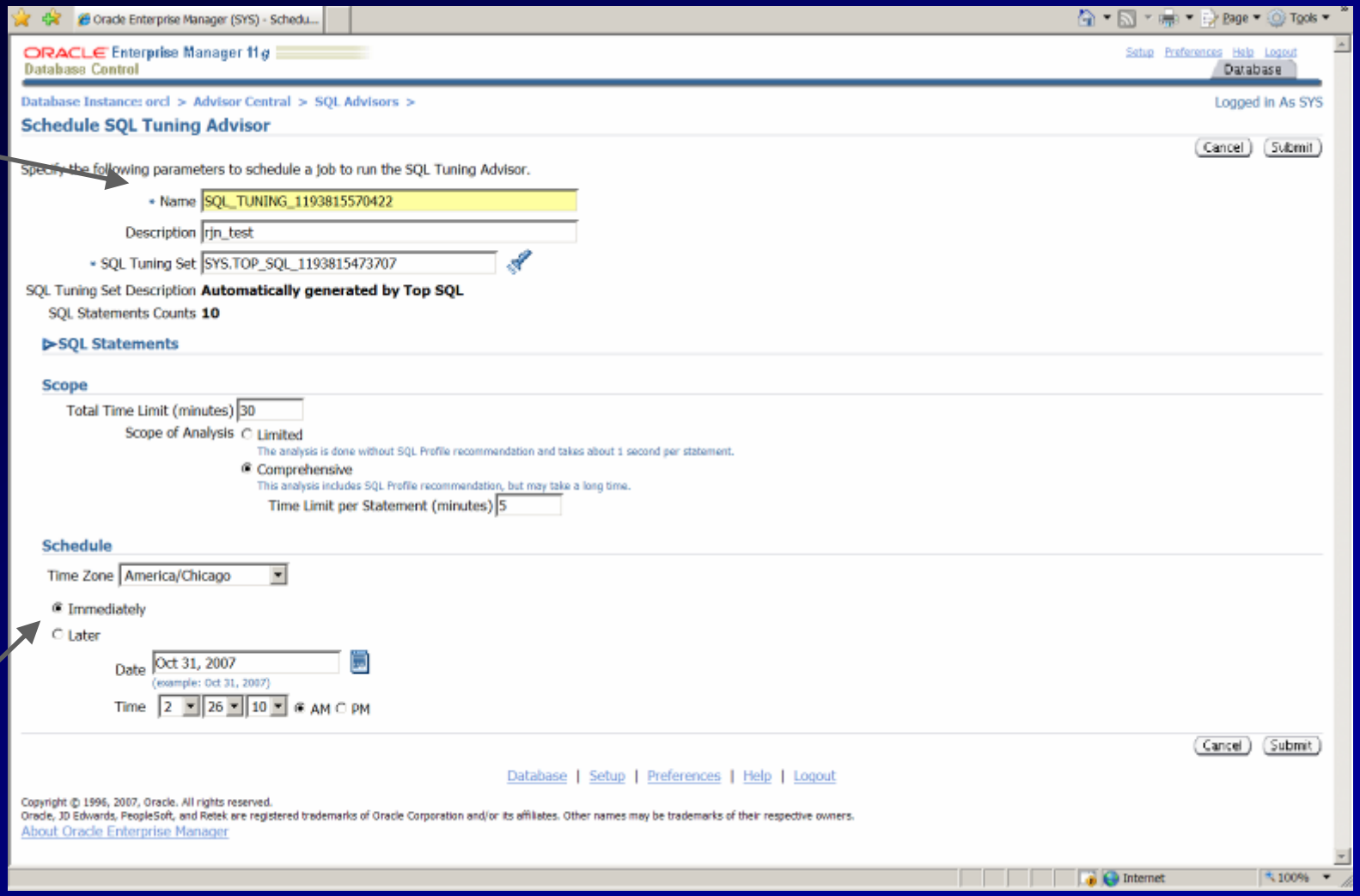
Queries & Stats



# SQL Plan Management
# Create a SQL Tuning Set

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

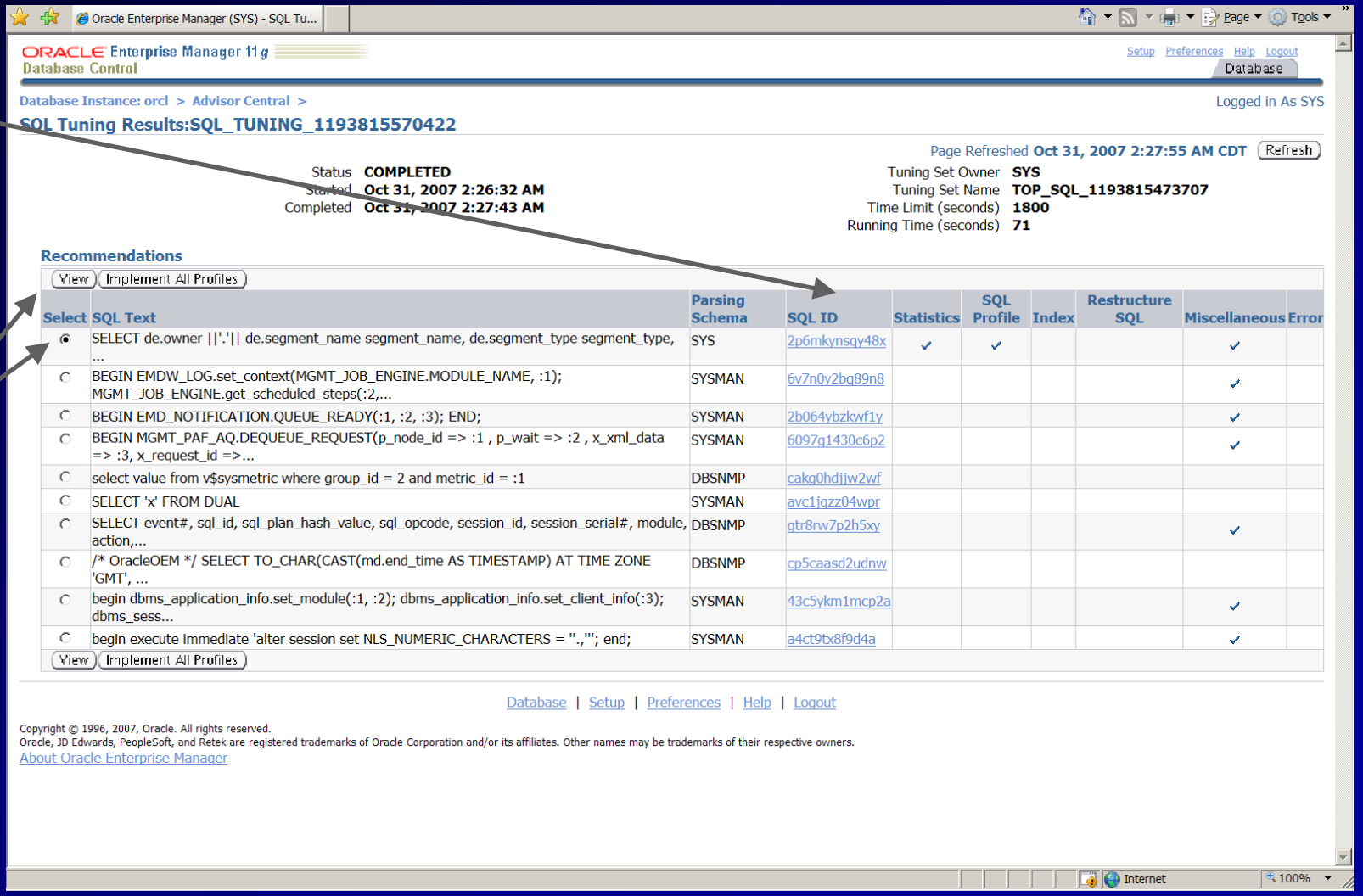
Run it NOW



# SQL Plan Management
# Create a SQL Tuning Set

Results

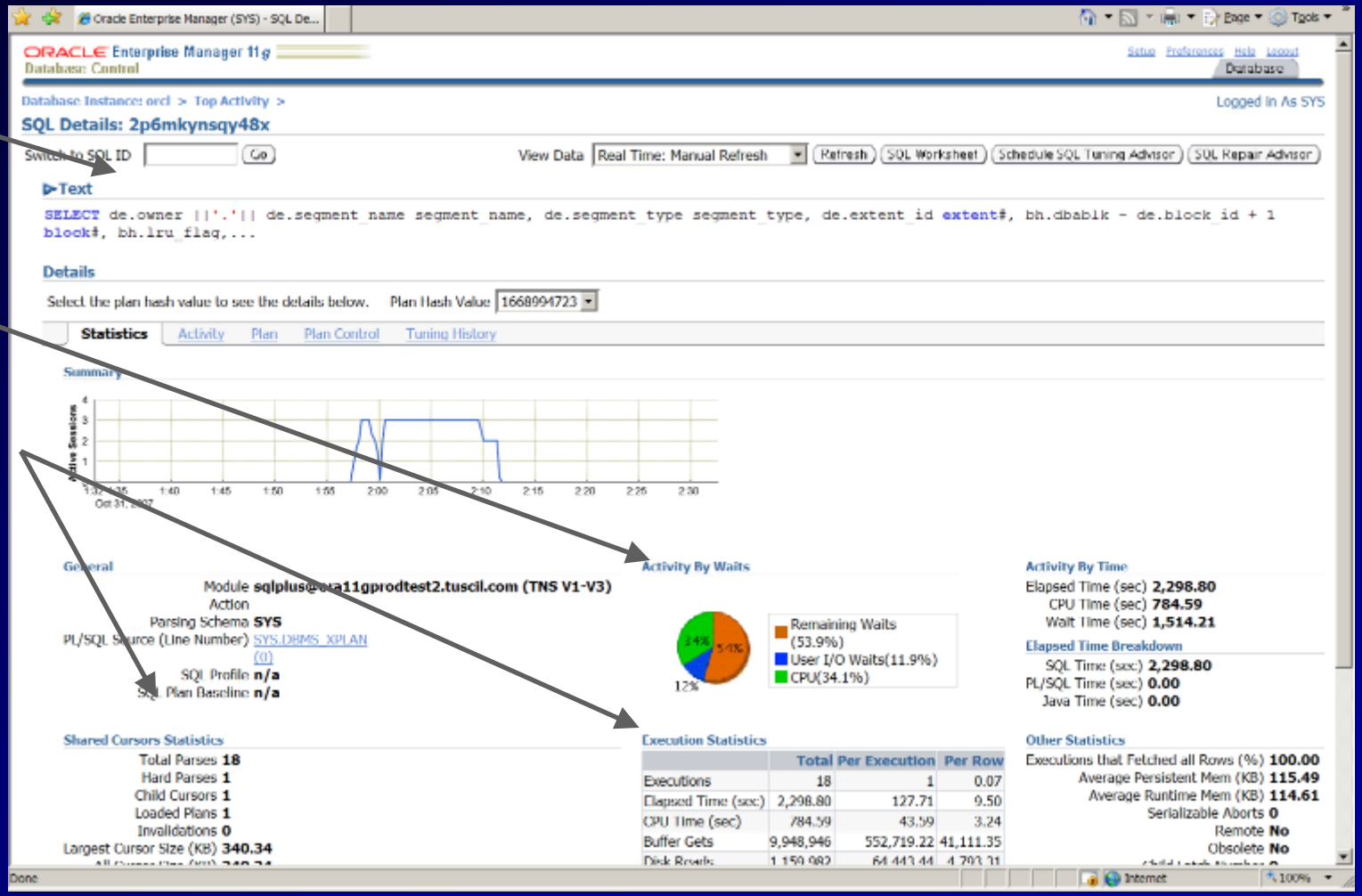
Select
One query
And click
View



# SQL Plan Management
# Click on any SQL ID

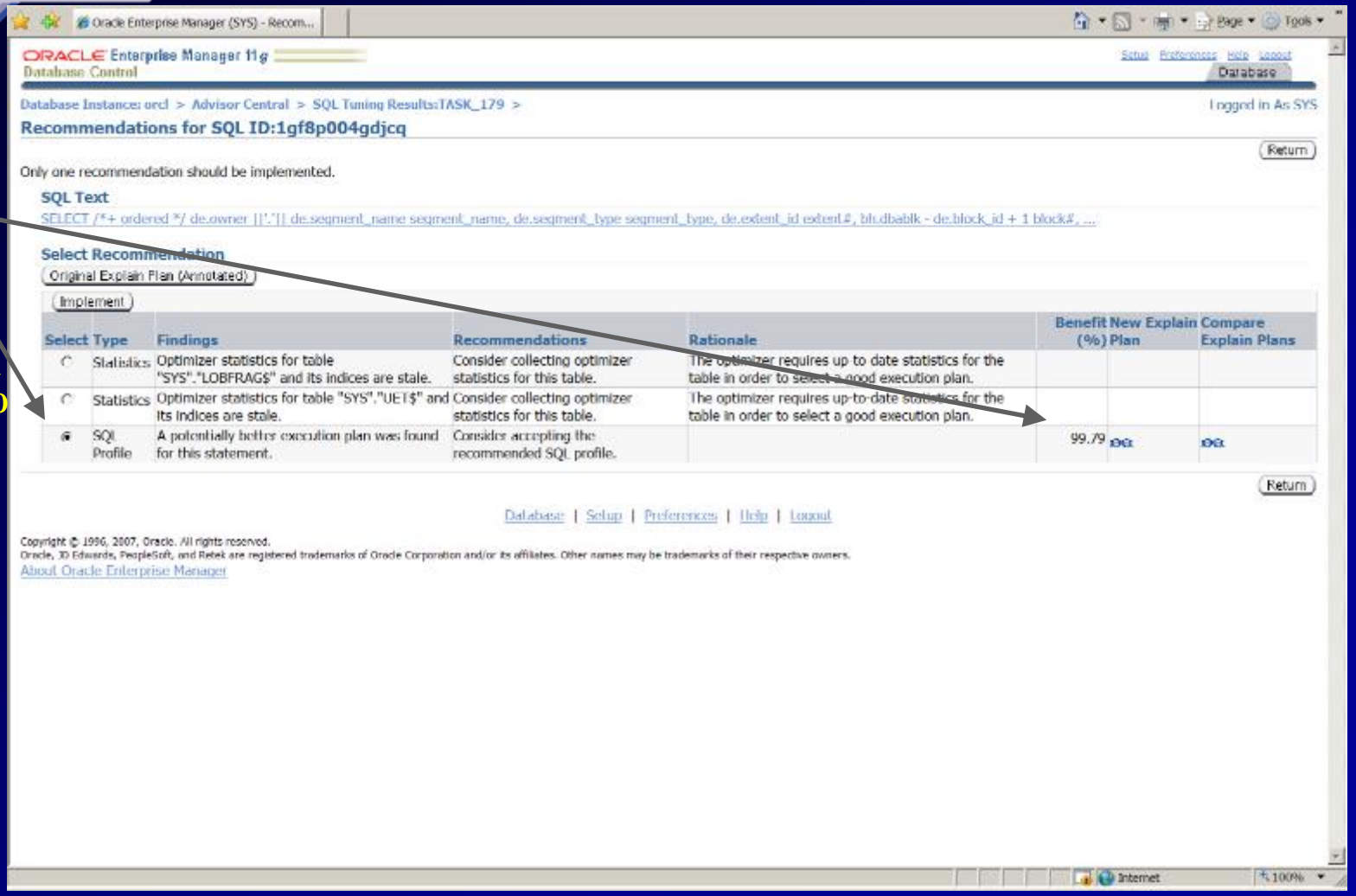
SQL Text

Waits & Statistics



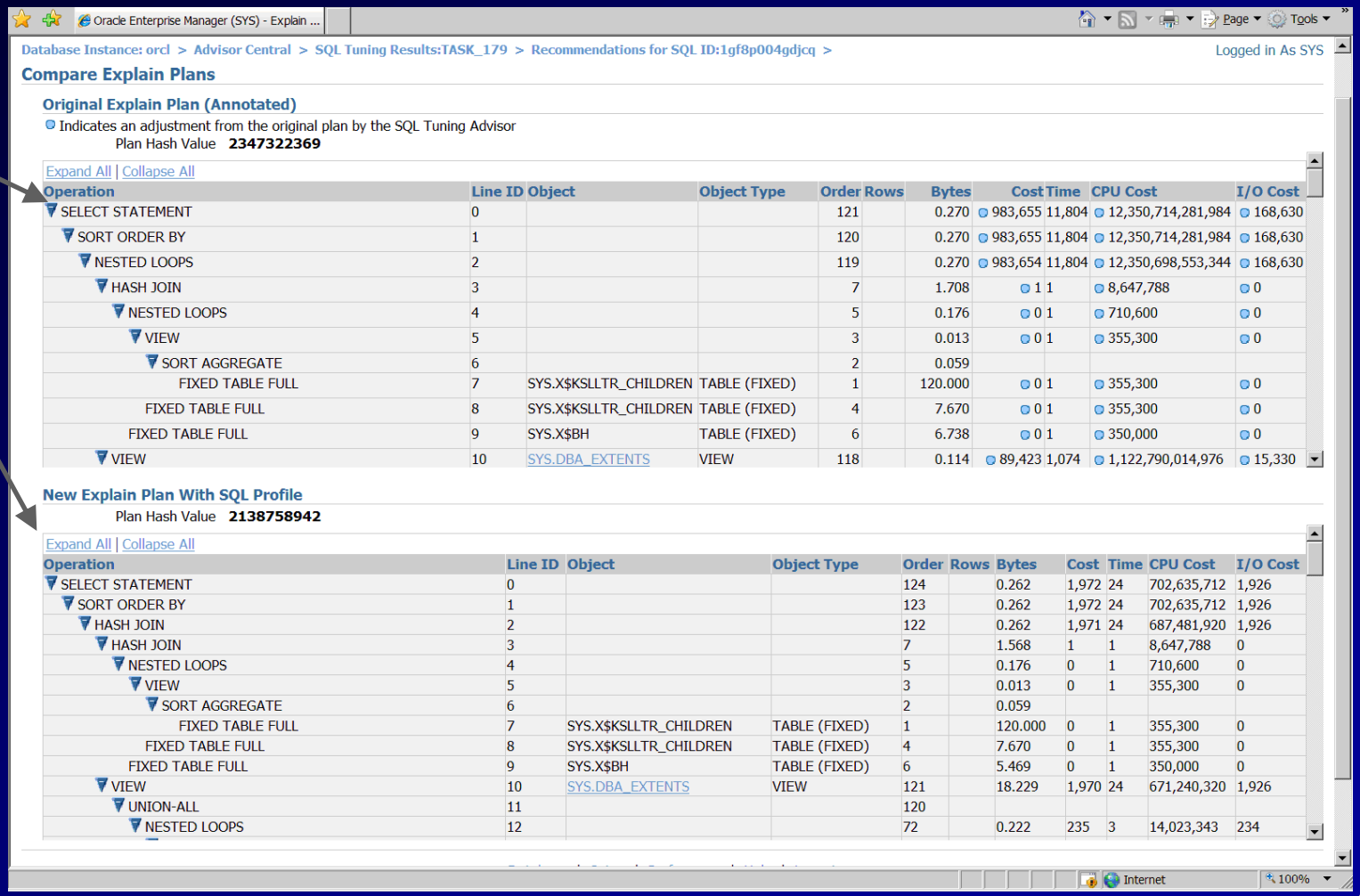
# SQL Plan Management
# Create a SQL Tuning Set

SQL Profile Will Help 99%



# SQL Plan Management
# Create a SQL Tuning Set

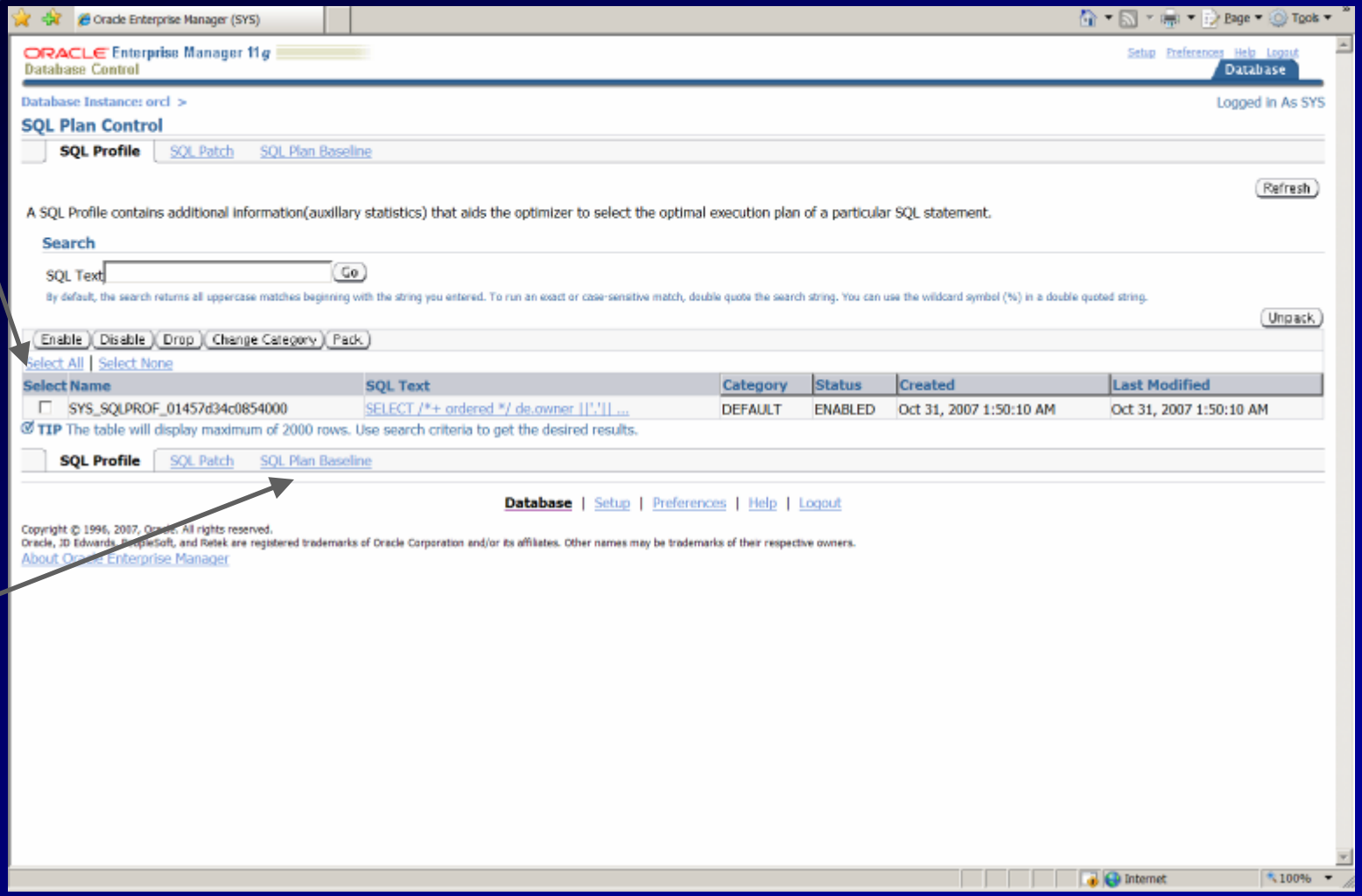
Compare Before & After



# SQL Plan Control
# SQL Profiles stored in the system

SQL Profiles

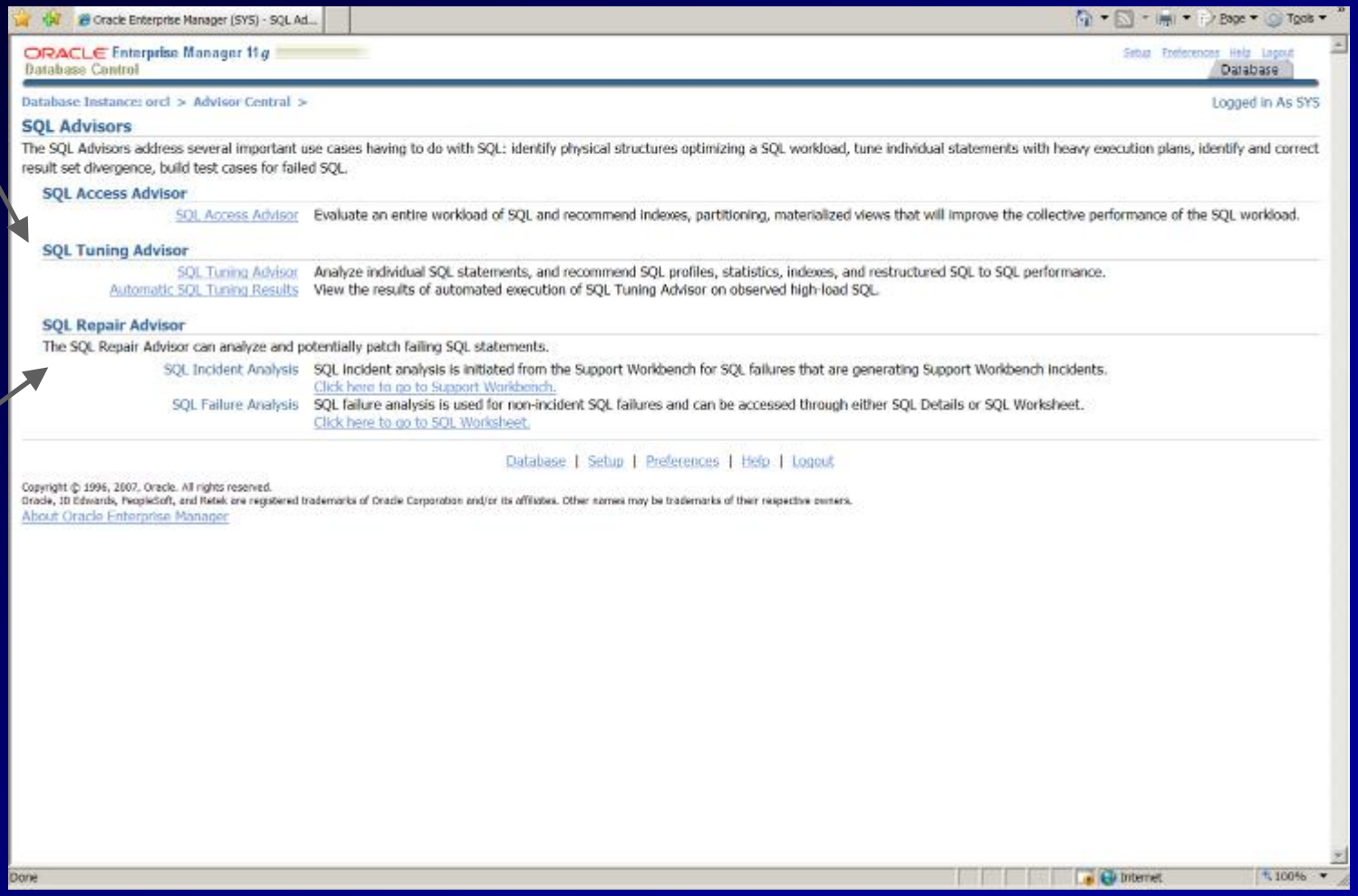
SQL Plan Baselines



# SQL Advisors

Tuning Advisors

Repair Advisor (next)



# The SQL Repair Advisor


ORA-600

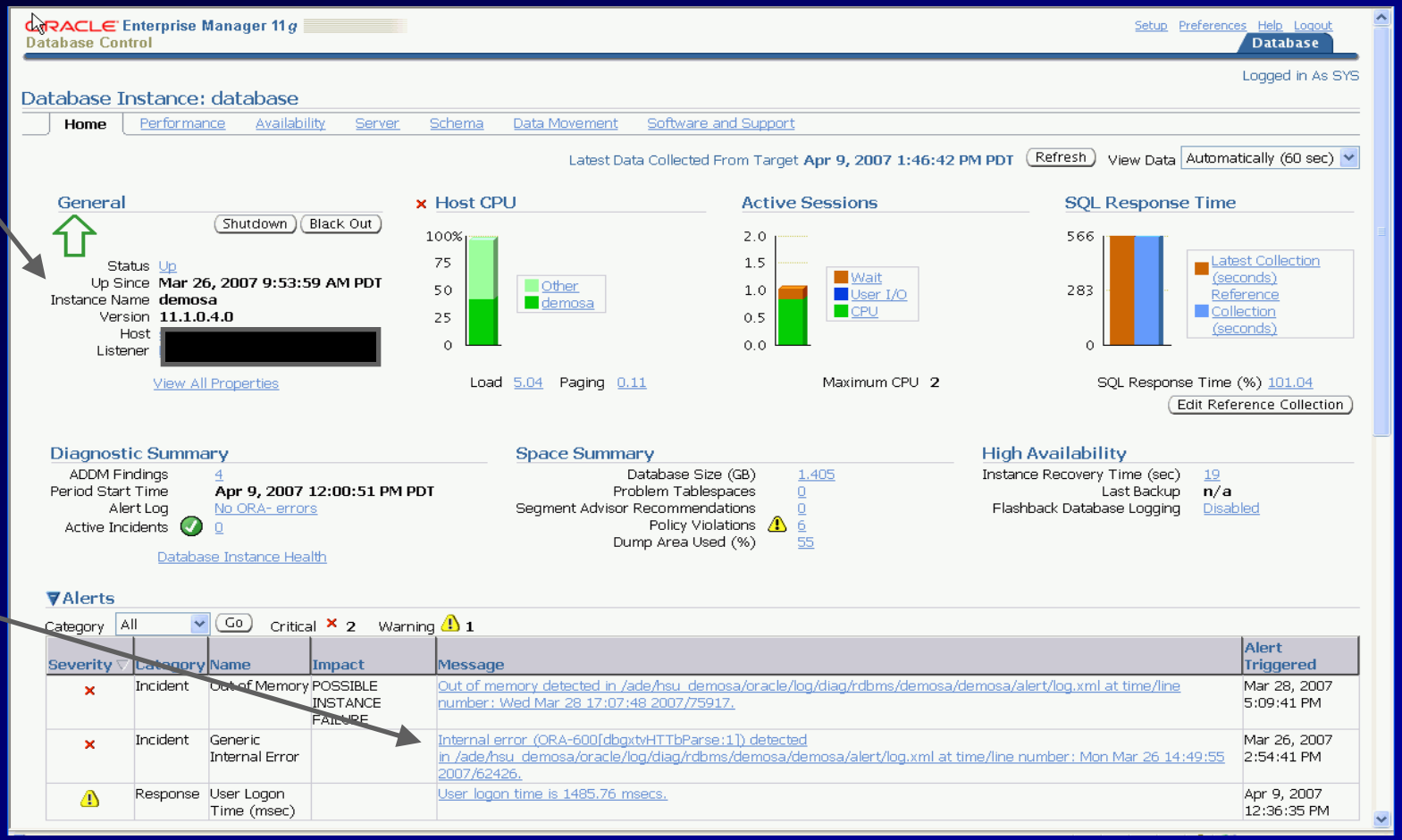
# SQL Repair Advisor

- 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

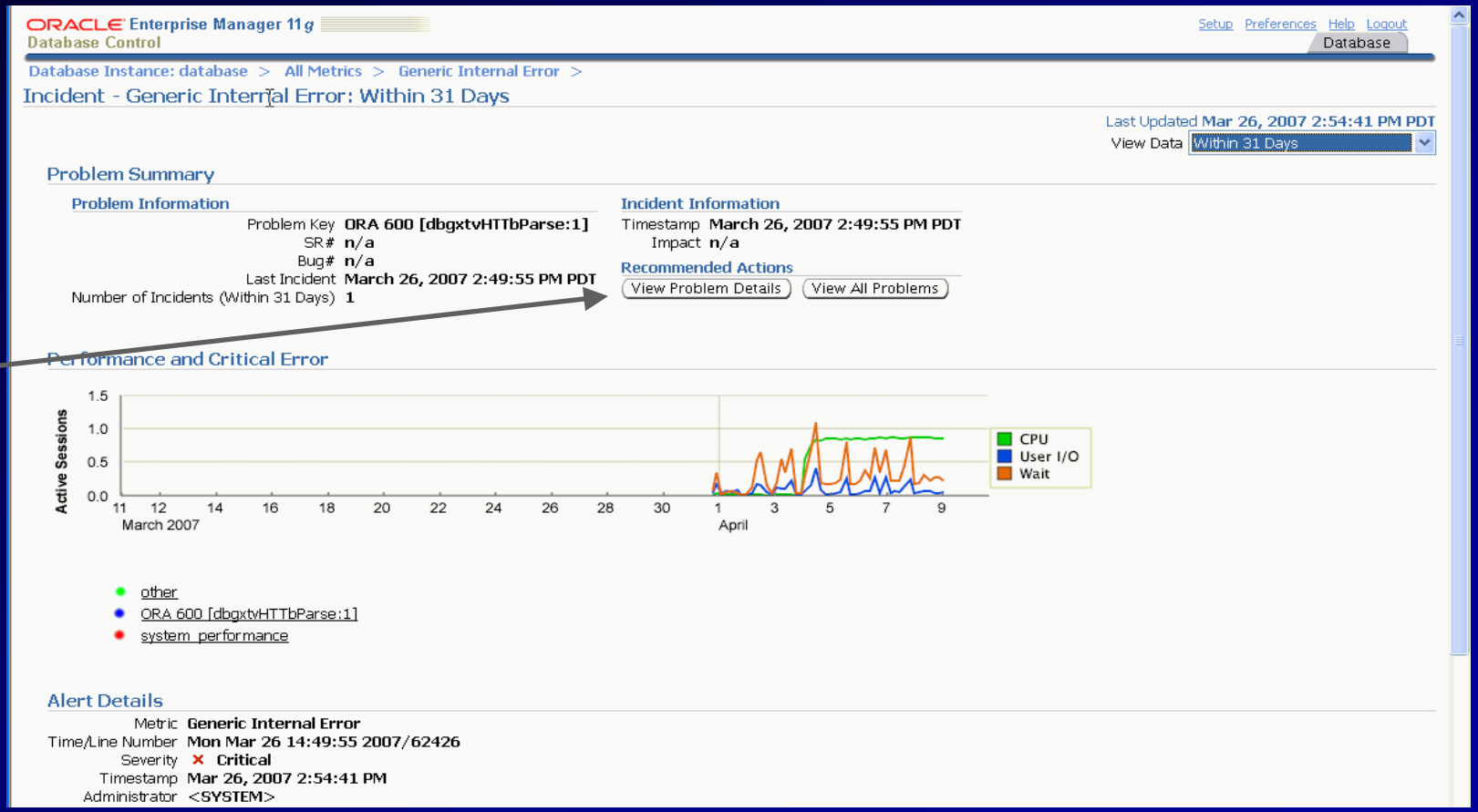
Go to the Database Instance

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



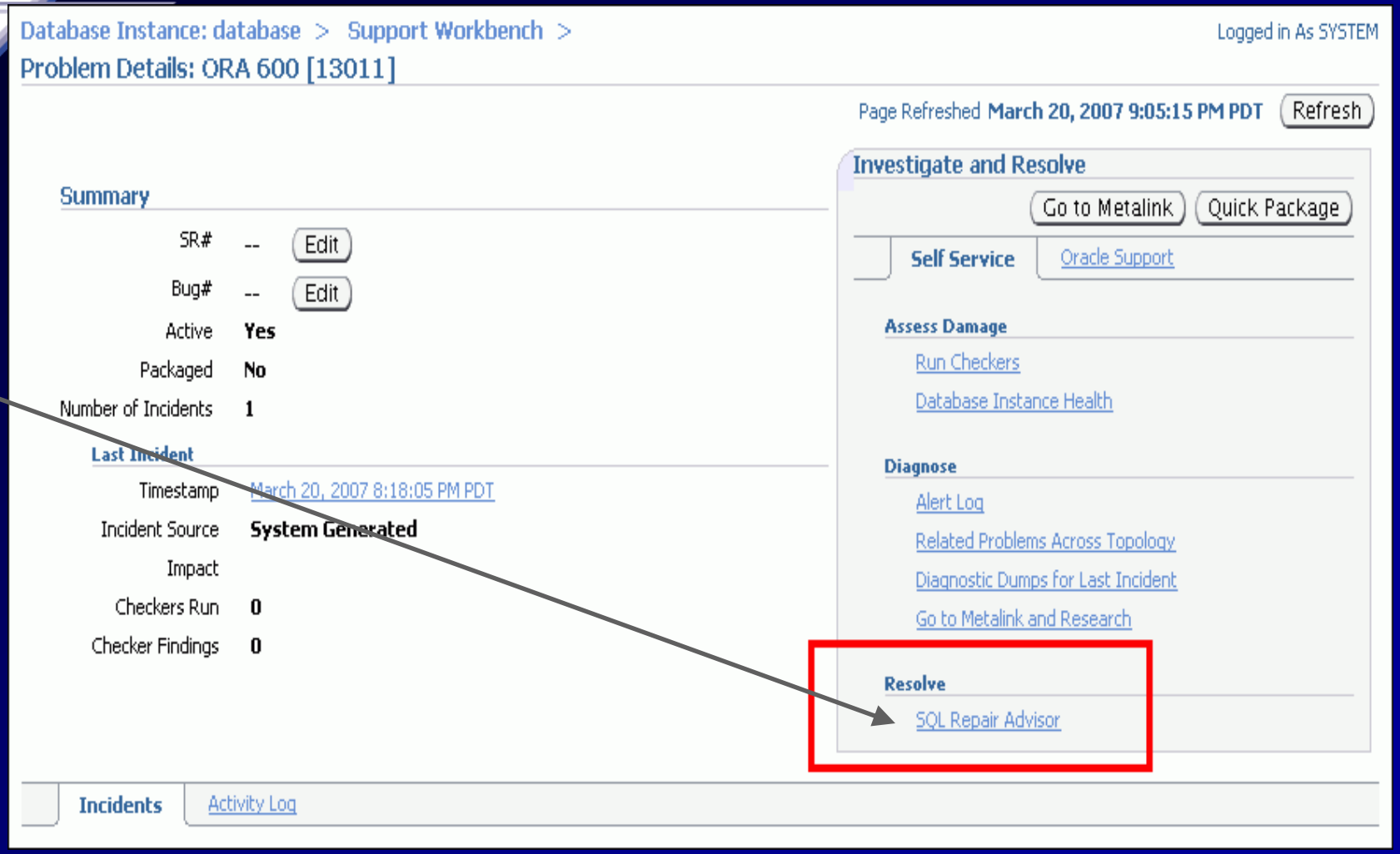
# SQL Repair Advisor – View Problem Details

Click on View Problem Details to go to the Support Bench



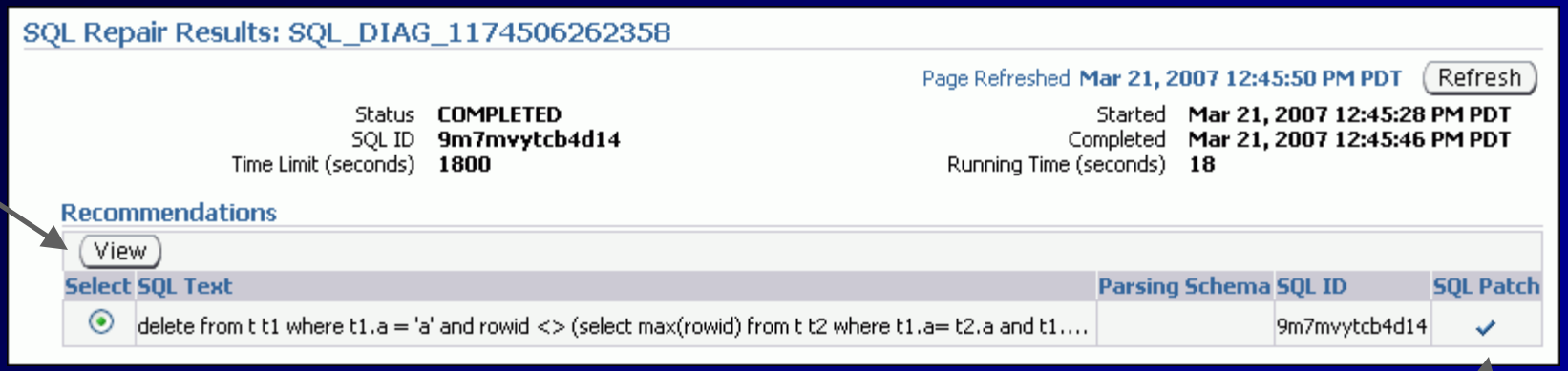
# Support Workbench - Details

Click on SQL Repair Advisor



# Results from SQL Repair Advisor

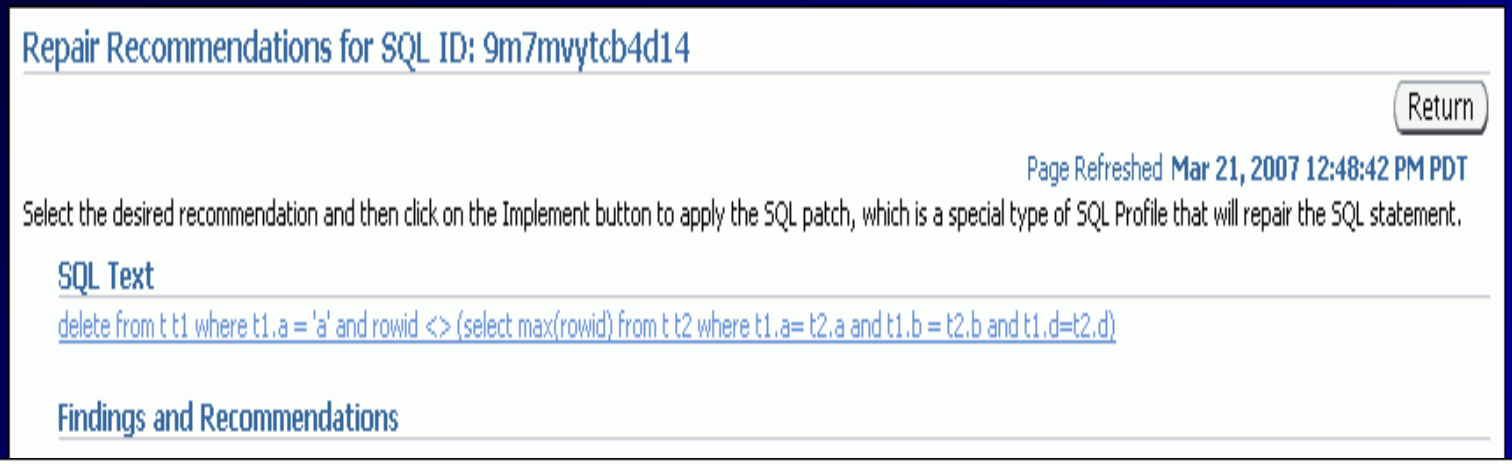
Click on View to Get the Detail finding of the Advisor

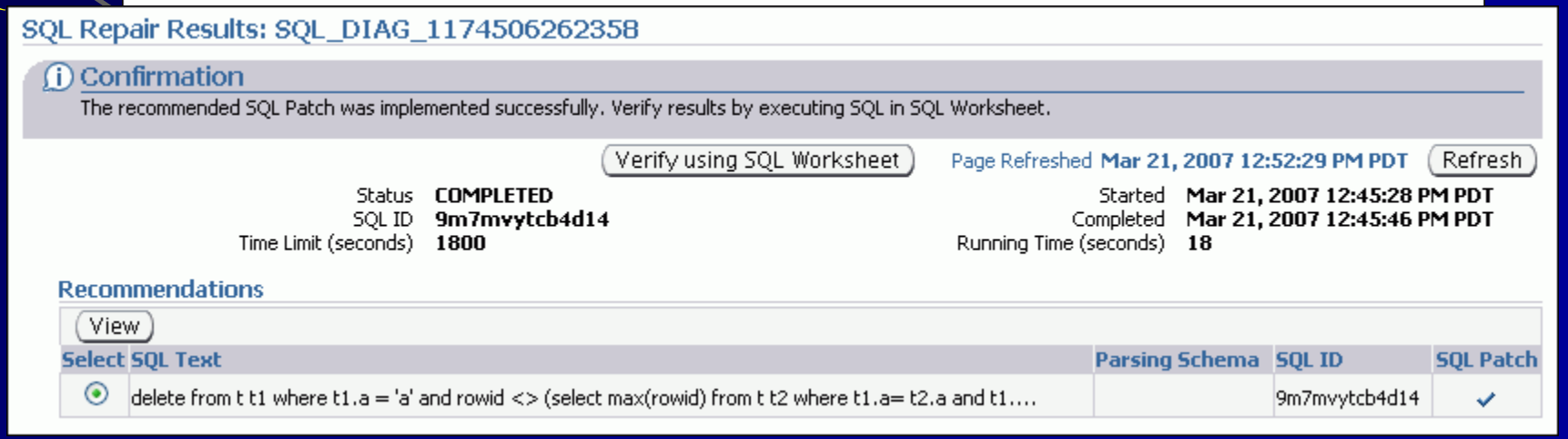



Note a SQL Patch (FIX for the SQL) has been generated

# SQL Repair Advisor
## Recommendation / Confirmation

Click on Implement To accept the SQL Patch



# SQL Performance Analyzer

# SQL Performance Analyzer

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

Great for:

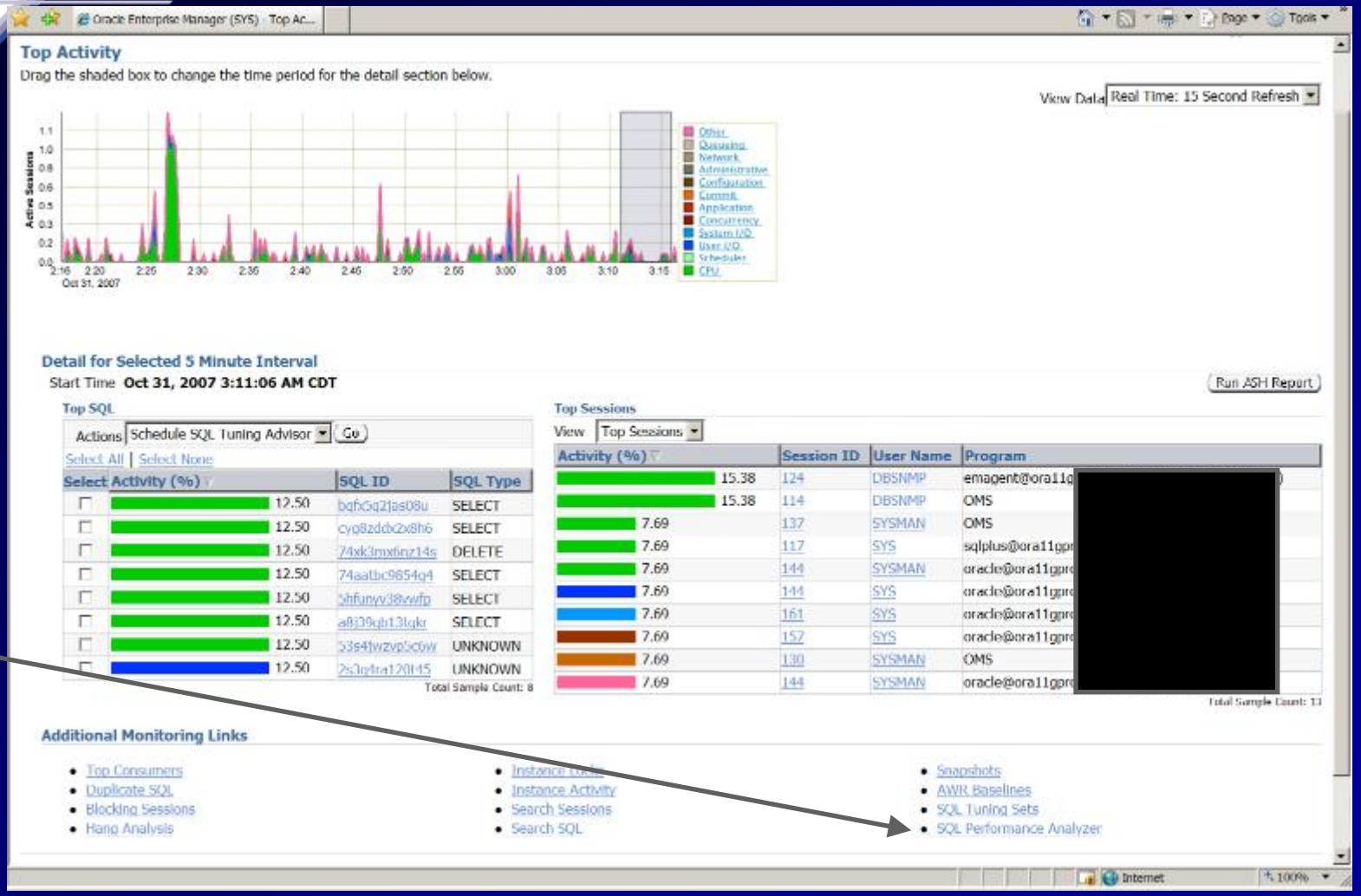
- Database Upgrades
- Application Upgrades
- Hardware Changes
- Database or Schema Changes
- Best for SQL Tuning – Especially Batches

# SQL Performance Analyzer

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

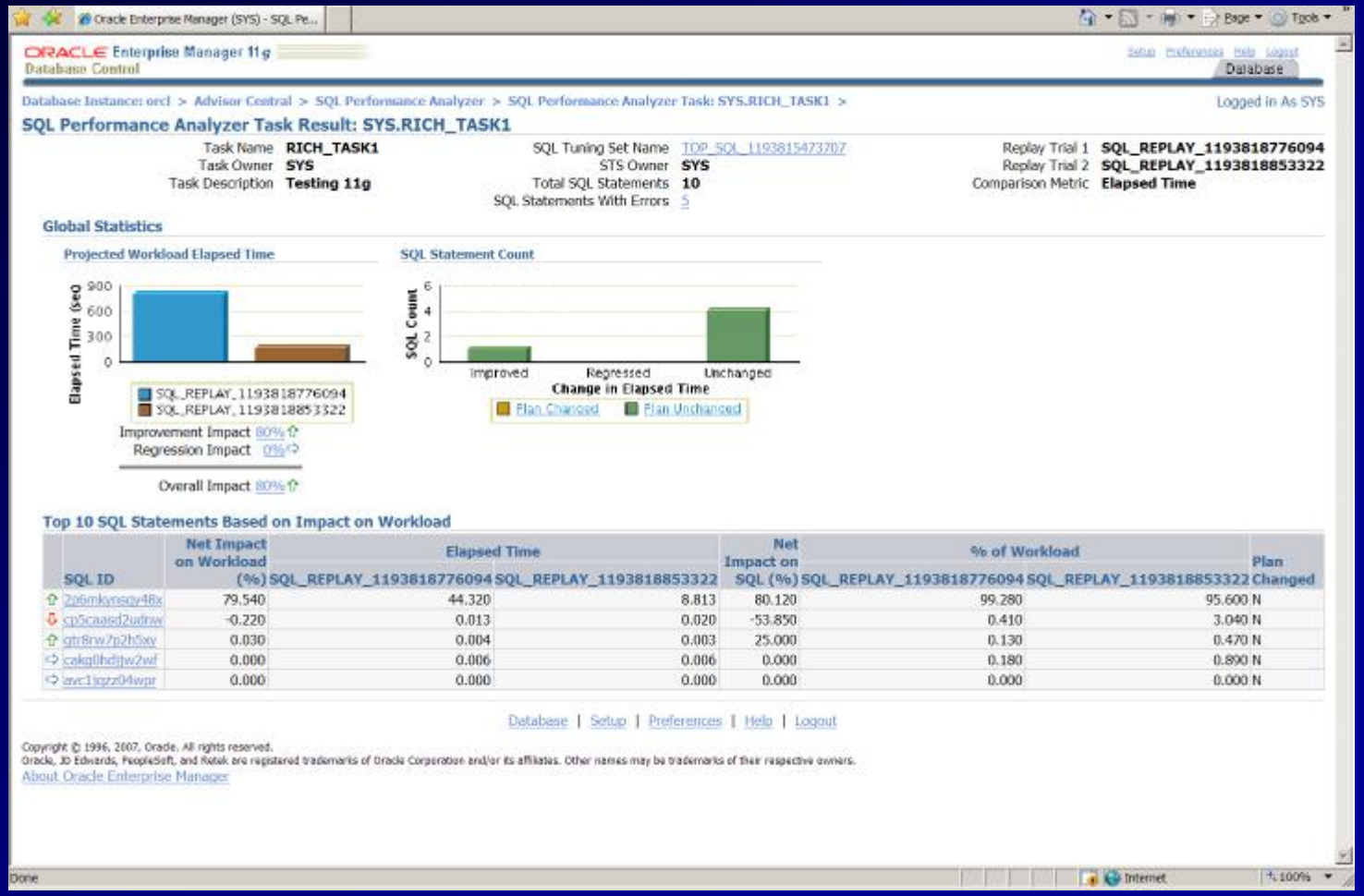
- Capture SQL
- Transport SQL
- Create a Replay Task
- Set up the environment to Test
- Make any changes to Test (such as SGA/Optimizer)
- Compare before and after performance
- Tune the problems!

# SQL Performance Analyzer

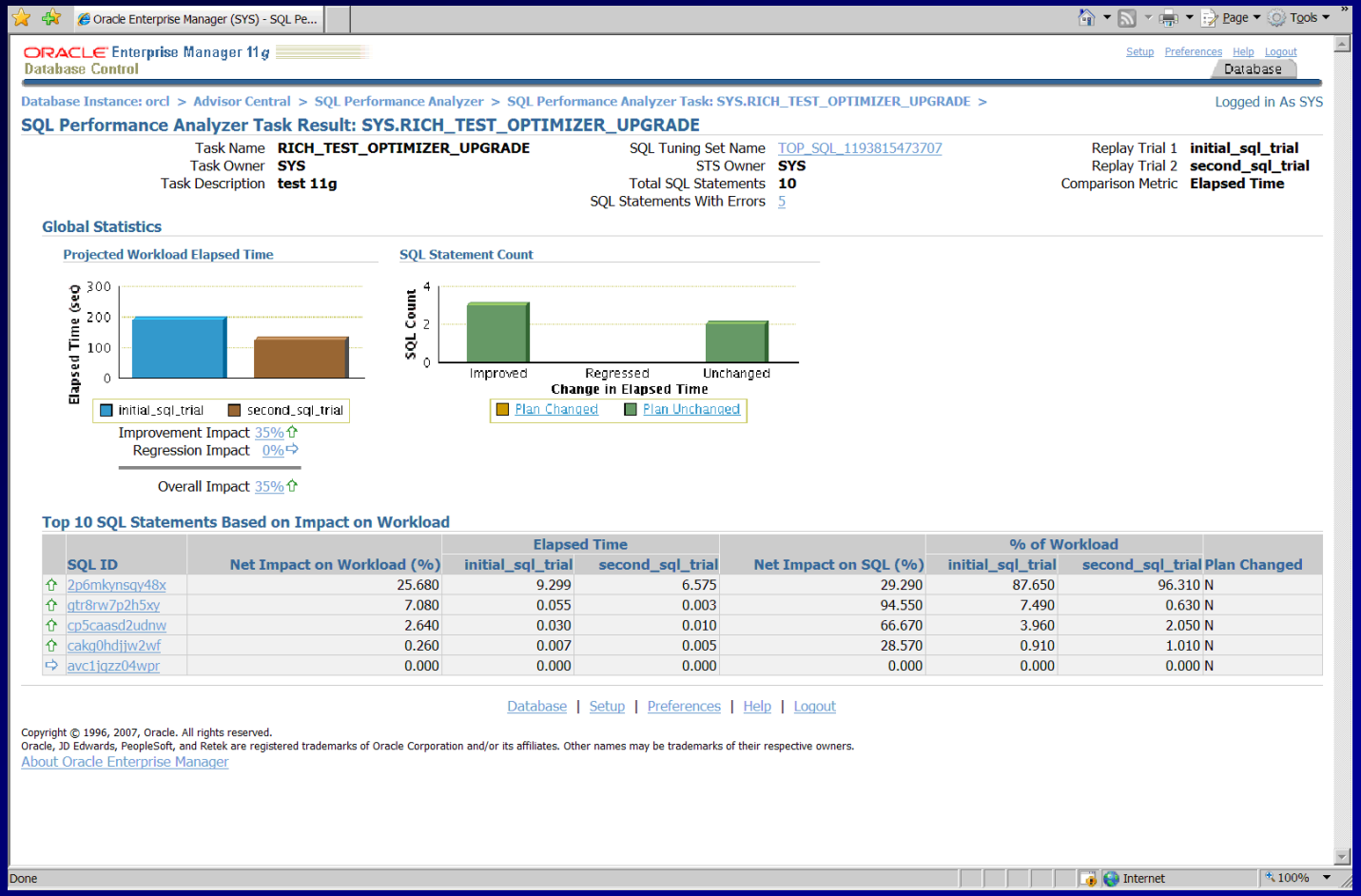



Click Here

# SQL Performance Analyzer Guided Workflow



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



# Real Application Testing!
# Database workload capture and replay

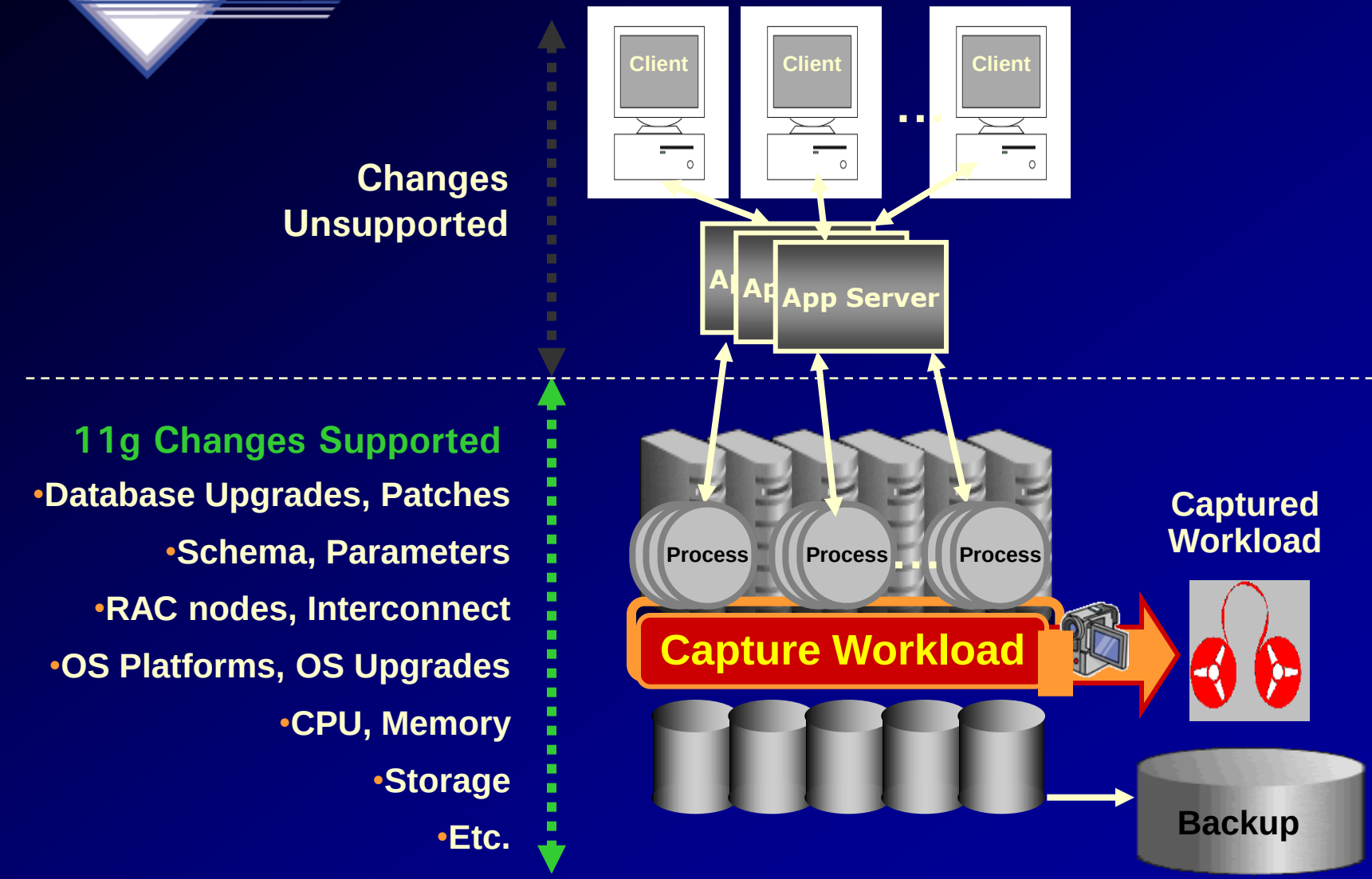

# Database workload capture and replay

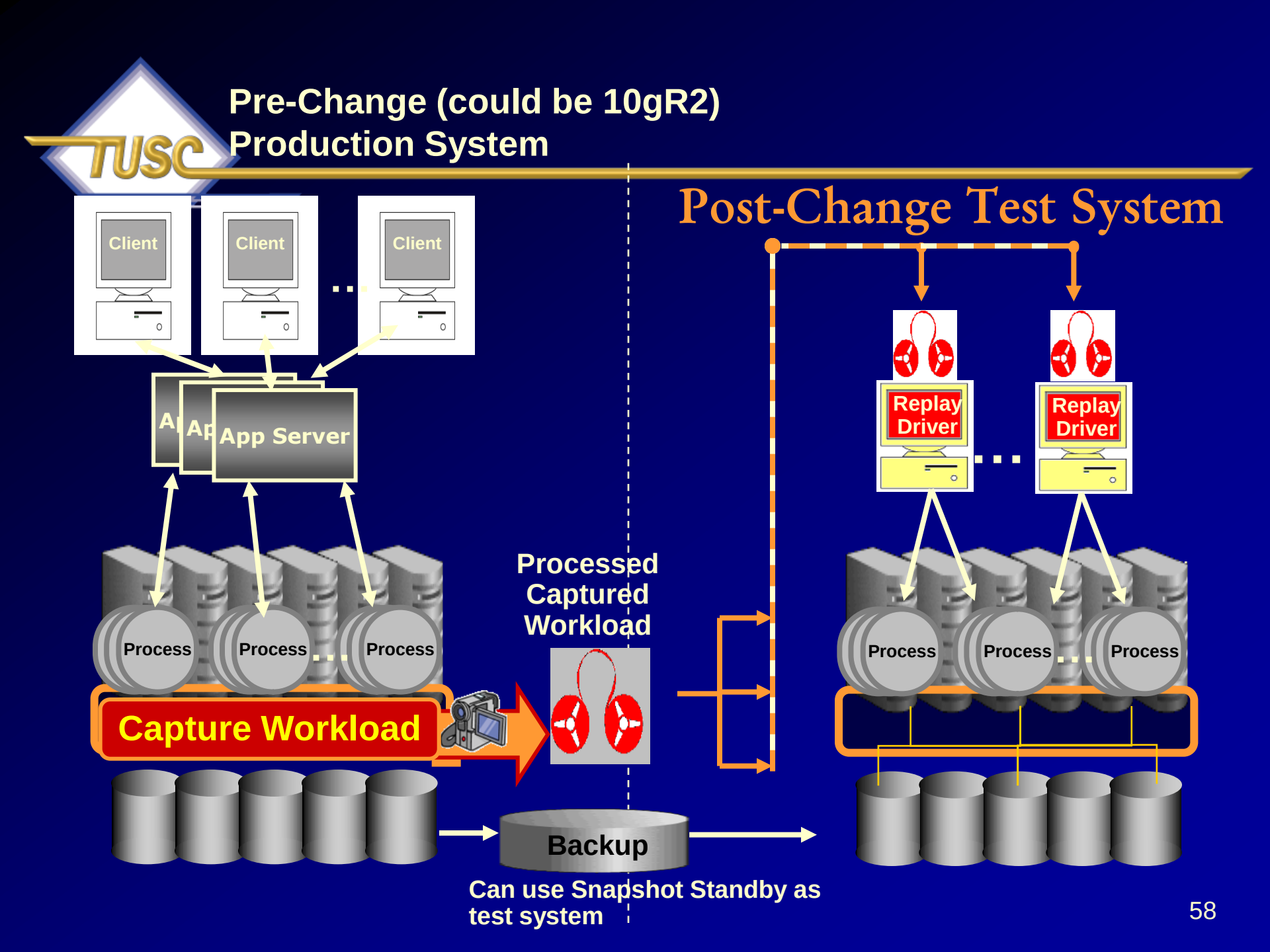
- Used to capture **database workload** on one system and replay later on a different system. Useful to compare two different systems.
- Could rival LoadRunner in the future (may be more precise!)

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)

# Pre-Change Production System

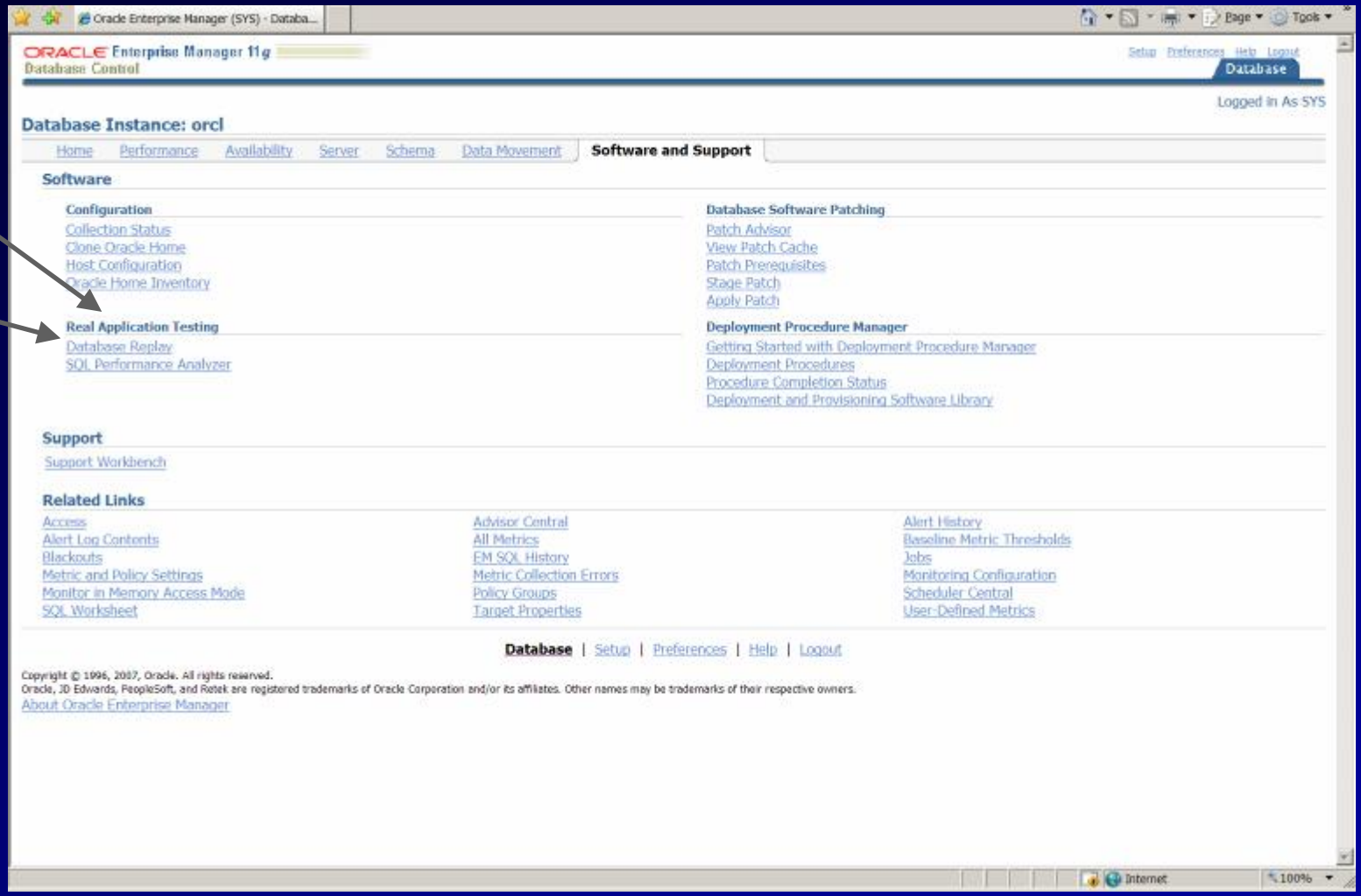



Pre-Change (could be 10gR2)
Production System
Post-Change Test System
Client
Client
Client
App Server
Process
Process
Process
Capture Workload
Processed
Captured
Workload
Backup
Replay
Driver
Replay
Driver
Process
Process
Process
Can use Snapshot Standby as
test system

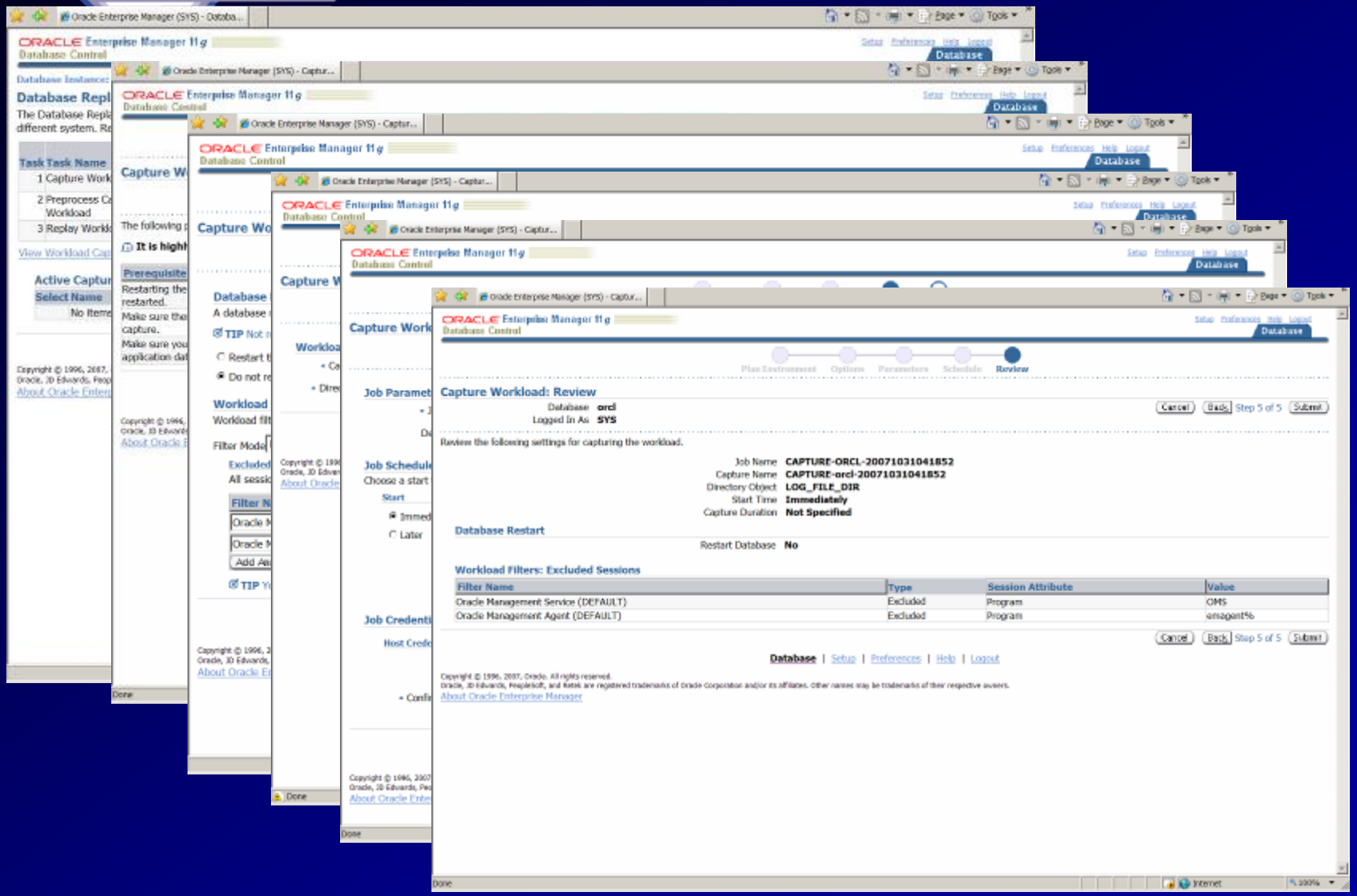
# Database Replay
# FYI Only – Download to view in detail

Real App Testing:

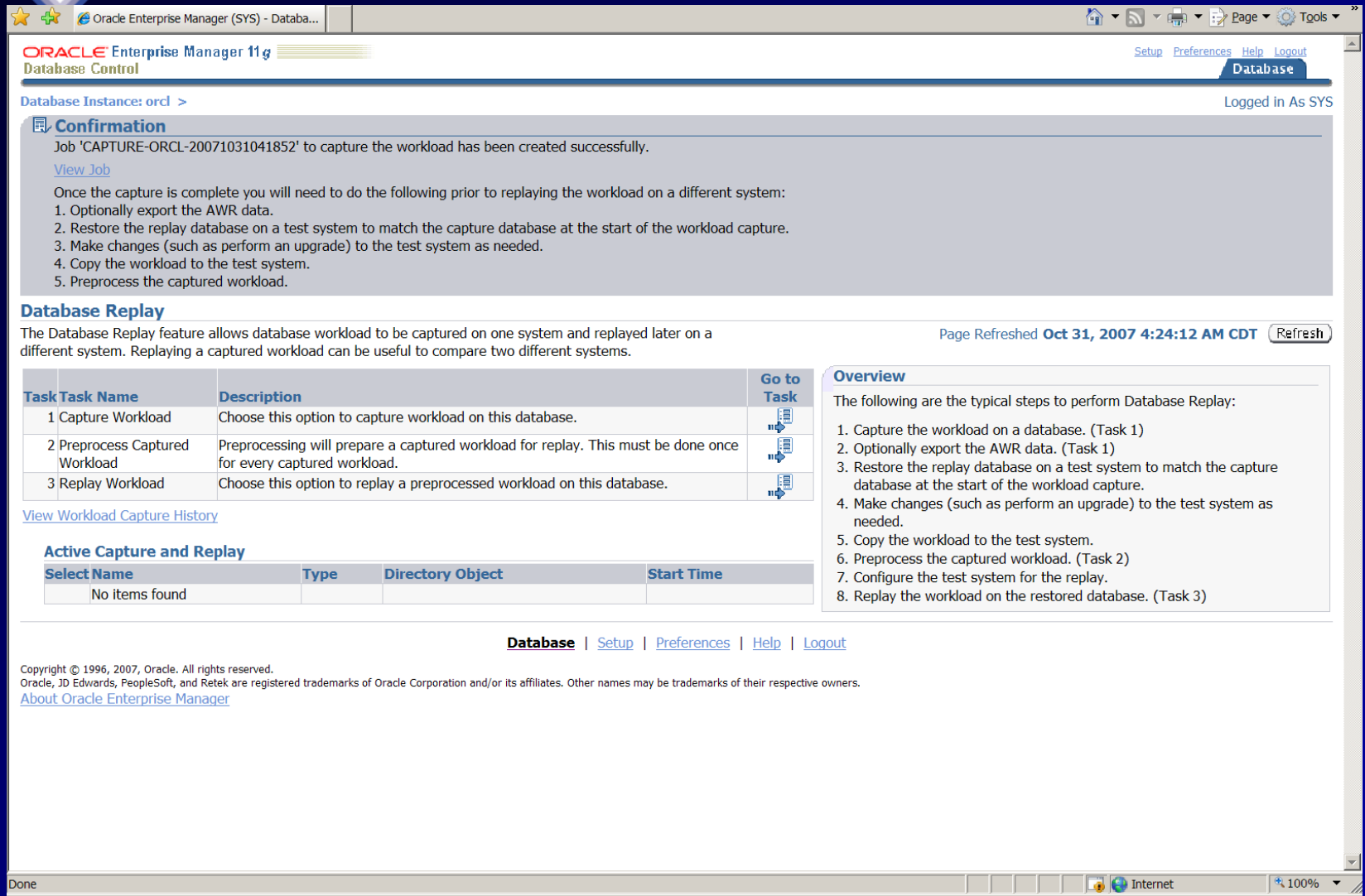
Database Replay



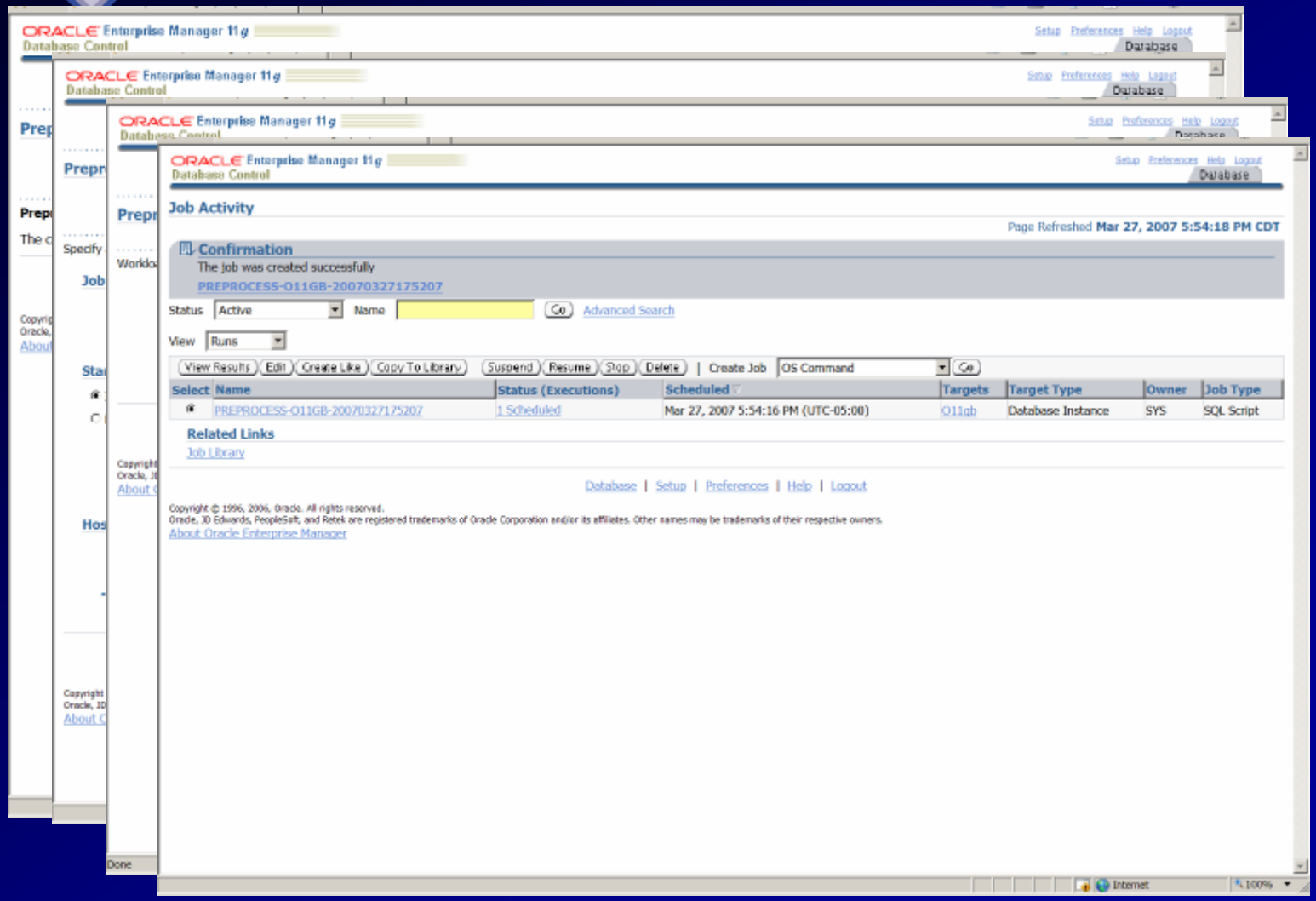
# Capture Workload – FYI Only



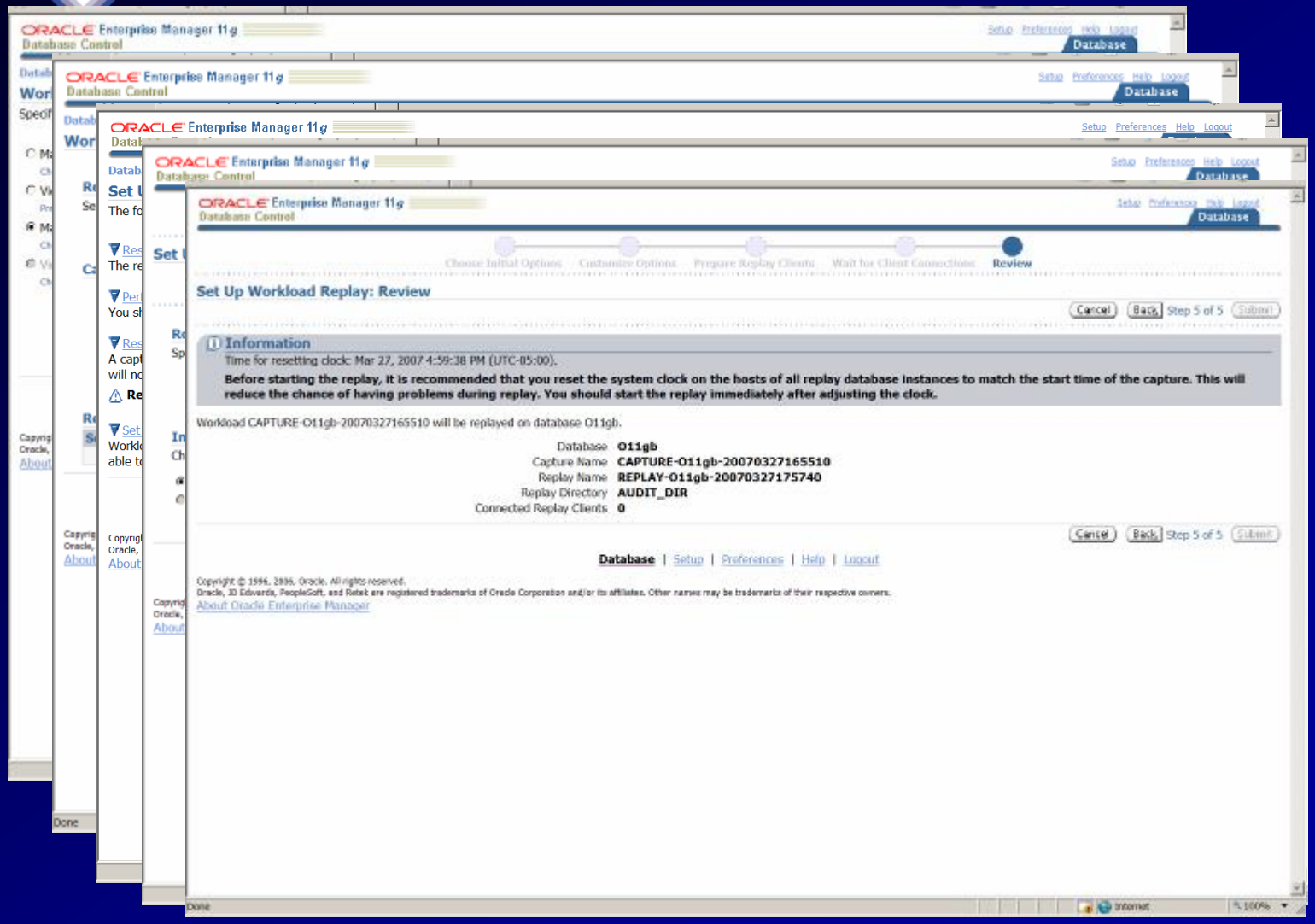
# Capture Workload – FYI Only



# Database workload – Preprocess FYI Only

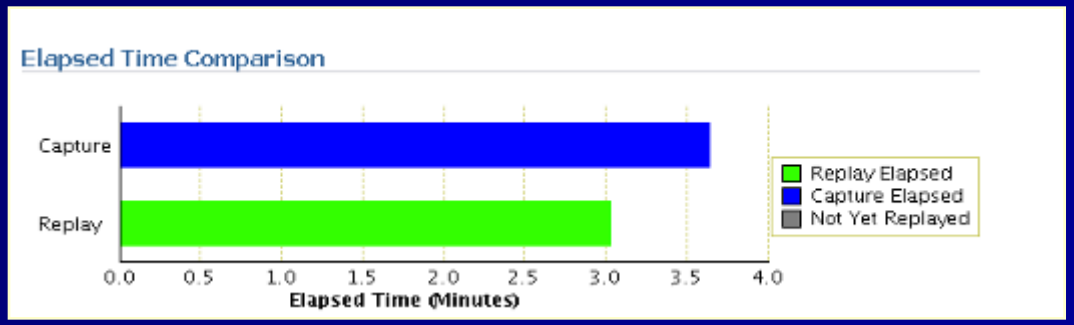


# Database workload capture and replay On NEW system (shortened) – FYI Only



# 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: (Briefly Only)

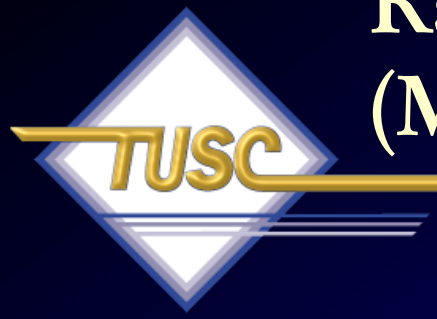
- Tables can be split into many pieces (10g).
- Only a subset of the data is queried
- All of the data COULD be queried
- Leads to enhanced performance of large tables
- Re-orgs & backups can be done on a partition level
- 4 quick examples follow (many many rules for each)
- WHAT'S NEW IN ORACLE 11G

# 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_yrinteger not null)
partition by hash (acct_no)
partitions 4
store in (users1, users2, users3, users4);
```

# Composite Partitioning v (8

```
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 aslo 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.
- 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
```


# 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

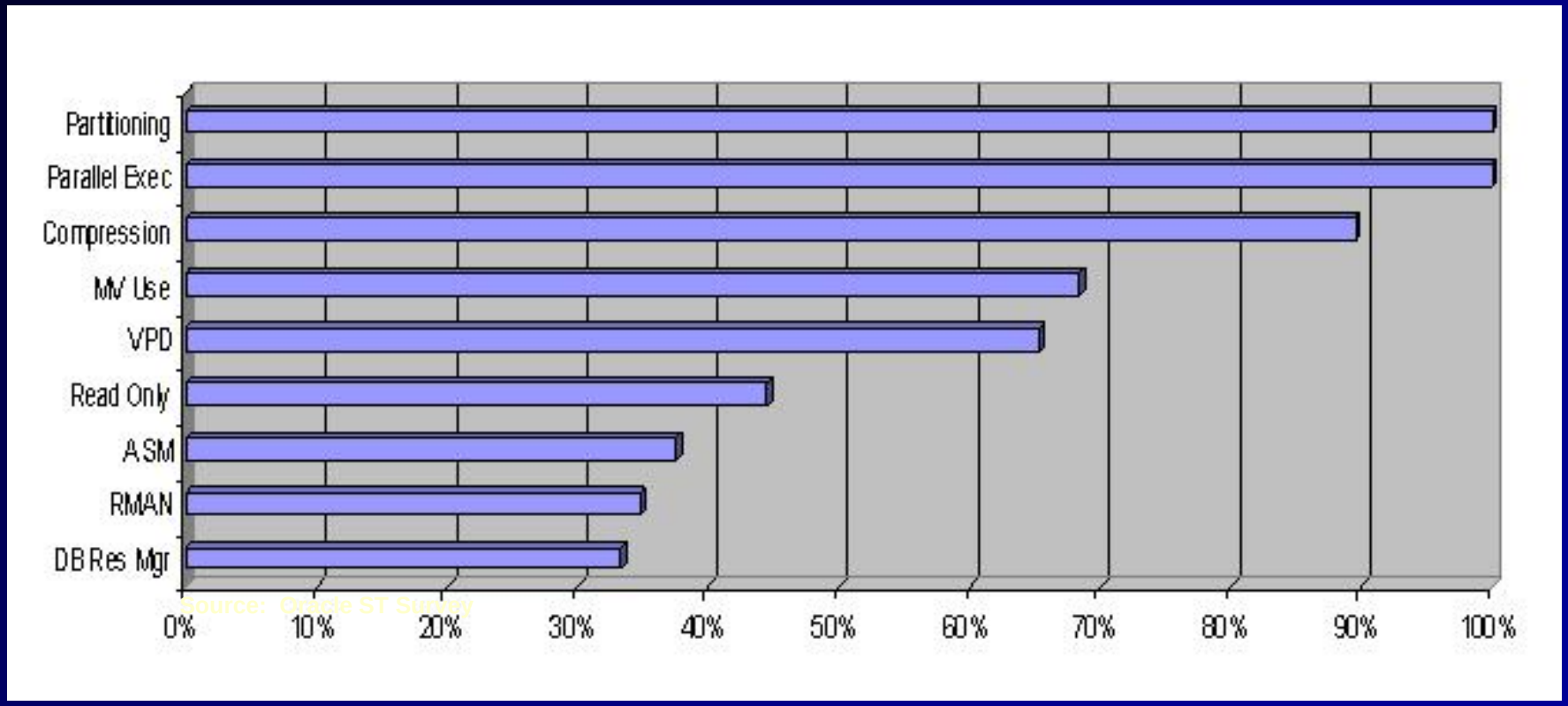
# Partition Compression

```
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.*

# Large-Scale Data Warehouses*

## *Feature Usage*




** Oracle Survey*

# Nice DBA Feature


# Oracle Secure Files
# FYI Only

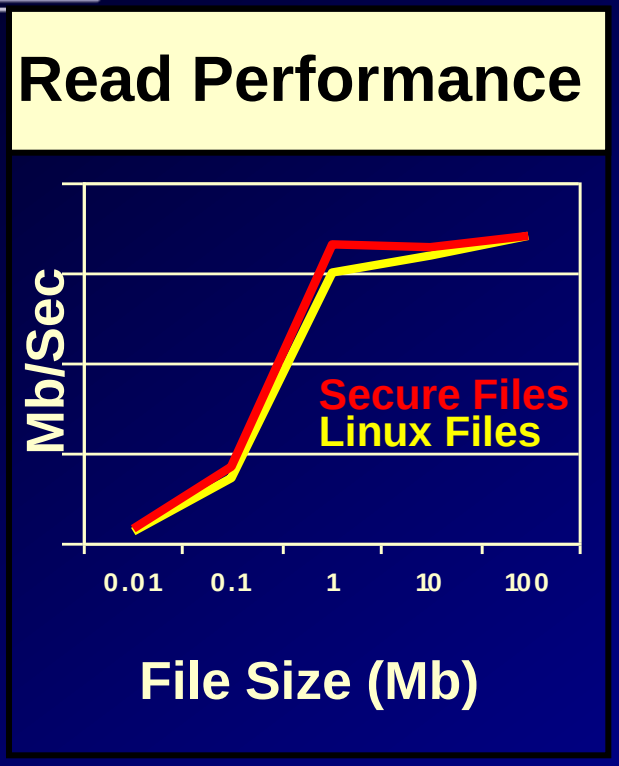
# Oracle SecureFiles

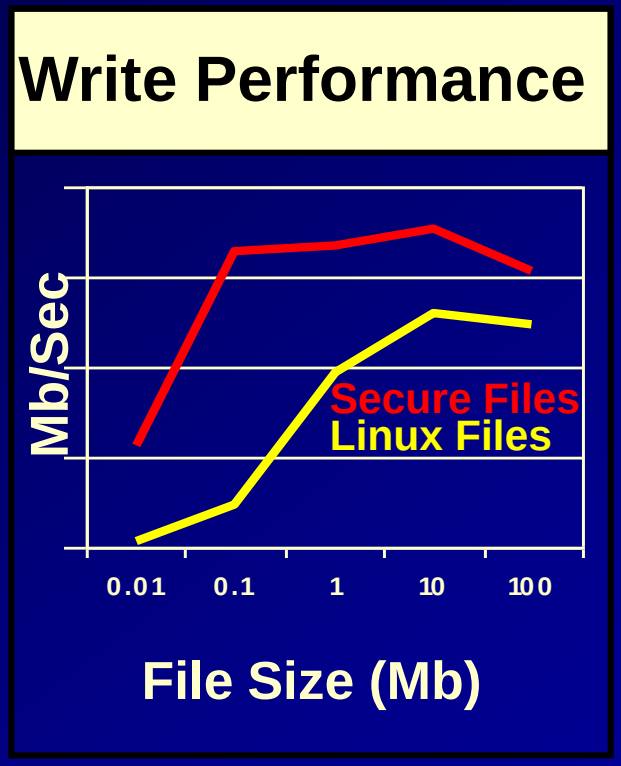
## *High-Performance Large Objects*

- 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....*

# Nice Developer Tools/Improvements


Result Cache
Invisible Indexes
PL/SQL Expressions

# 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** (**manual**) – no auto
- 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 with the RESULT_CACHE hint and it returned an elapsed time of 3.18 seconds (cache it).

Step 2 - In Session 2

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

# The Result Cache – Autotrace

select count(*) from emp; (note the result cache)

```
COUNT(*)
----------
        14

Execution Plan
----------------------------------------------------------
Plan hash value: 2937609675

-------------------------------------------------------------------------------------
| Id  | Operation          | Name                         | Rows  | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |                              |     1 |     1   (0)| 00:00:01 |
|   1 |  RESULT CACHE      | 4ntcq5q3m4ayb26wqthu7pbn17   |       |            |          |
|   2 |   SORT AGGREGATE   |                              |     1 |            |          |
|   3 |    INDEX FULL SCAN | PK_EMP                       |    14 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------
```


# The Invisible Index

ORACLE 11g DATABASE

- 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 NO_INDEX (to override visibility).
- The index IS MAINTAINED during DML
- Great for testing!

# 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;
```

# Create & Rebuild Index Online


# Create & Rebuild Index Online

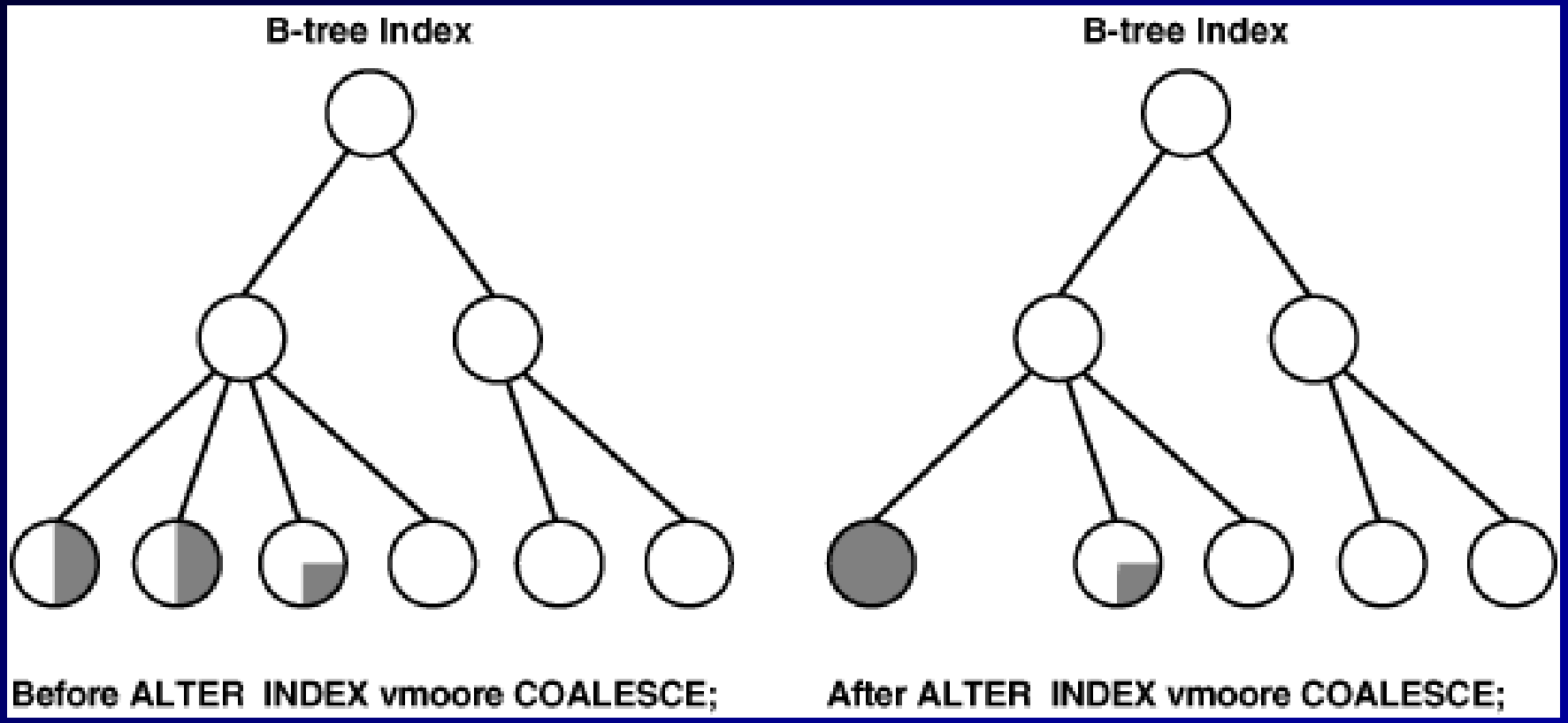
ORACLE 11g DATABASE

- 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


ORACLE 11g DATABASE

Rebuild:

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

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

# 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

# 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!

## 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
```

# 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
```

# 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

# Example

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

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

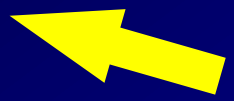
- Three records selected.
- Single column statistics are accurate

| 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 |



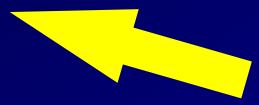
| Make | Price | Color |
|---|---|---|
| CORVETTE | 40,000 | RED |
| CORVETTE | 60,000 | BLACK |
| CORVETTE | 50,000 | SILVER |
| CADILLAC | 90,000 | RED |
| JEEP | 35,000 | BLACK |
| JEEP | 45,000 | SLIVER |

- 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 |



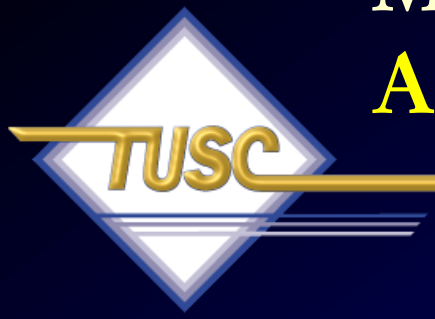
| Make | Price | Color |
|---|---|---|
| CORVETTE | 50,000 | RED |
| CORVETTE | 50,000 | BLACK |
| CORVETTE | 50,000 | SILVER |
| CADILLAC | 90,000 | RED |
| JEEP | 35,000 | BLACK |
| JEEP | 45,000 | SLIVER |

- 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';

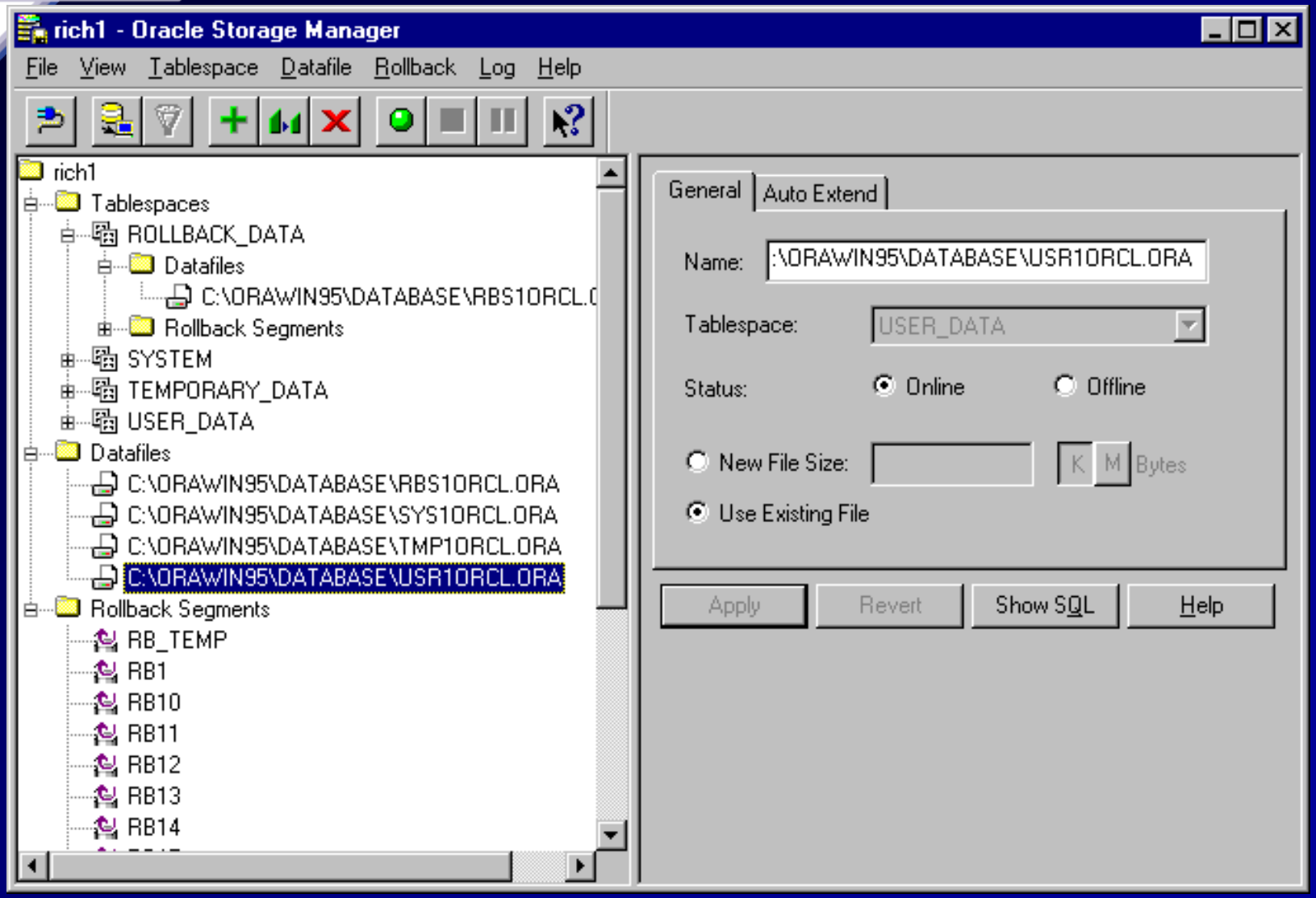
```
COLUMN_NAME                    NUM_DISTINCT HISTOGRAM
------------------------------ ------------ ---------------
SYS_STUJGVLRVH5USVDU$XNV4_IR#4          145 FREQUENCY
CUST_VALID                                2 FREQUENCY
COUNTRY_ID                               19 FREQUENCY
CUST_STATE_PROVINCE                     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.
```


# Enterprise Manager for the Grid

# Enterprise Manager: Back in Time!



# Performance Manager : Back in Time!


OVERVIEW - rich1

Window Refresh Help

Automatic Refresh Interval: 0 min 30 sec

BUFFER CACHE HIT%
4.14%
95.86%
Hit % Miss %

LIBRARY CACHE HIT% ...
99.91%
Hit % Miss %

DATA DICT CACHE HIT%
12.41%
87.59%
Hit % Miss %

MEMORY SORT HIT%
1.89%
98.11%
Hit % Miss %

ROLLBACK NOWAIT HIT%
100%
Hit % Miss %

USERS LOGGED ON
11.00
11.00 8.80 6.60 4.40 2.20 0.00
1

ACTIVE USERS
5.00
6.00 4.80 3.60 2.40 1.20 0.00
1

USERS RUNNING
4.00
4.00 3.20 2.40 1.60 0.80 0.00
1

USERS WAITING FOR LOCKS
0.00
1.00 0.80 0.60 0.40 0.20 0.00
1

MEMORY ALLOCATED ...
688320
688320 550656 412992 275328 137664 0
1

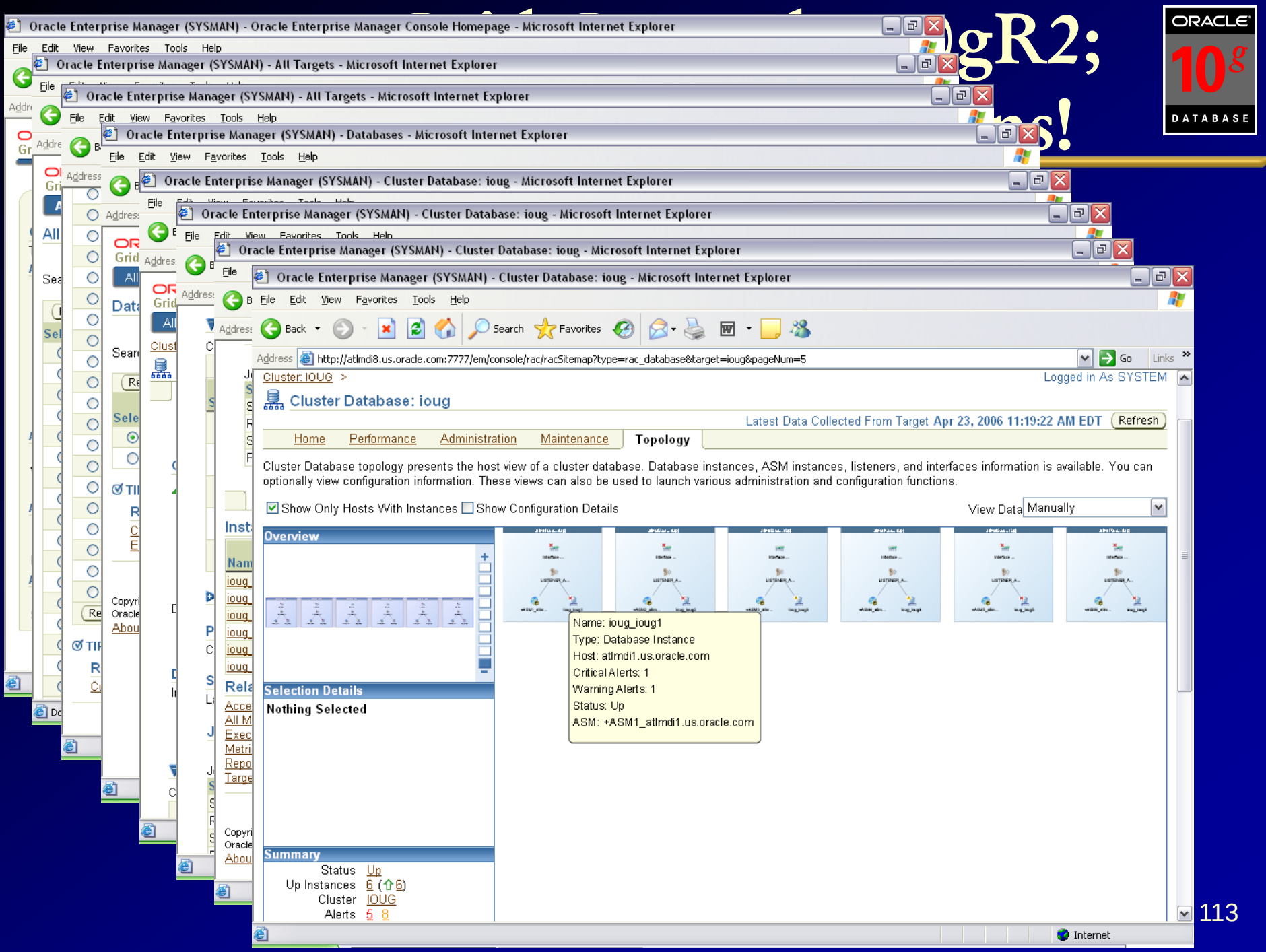
FILE I/O RATE ...
44.00 22.00 0.00
Physical Reads/s Physical Writes/s

SYSTEM I/O RATE
590 295 0
Buffer Gets/s Block Changes/s

THROUGHPUT ...
8.83 0.00
User Calls/s Transactions/s

For Help, press F1

# gR2;

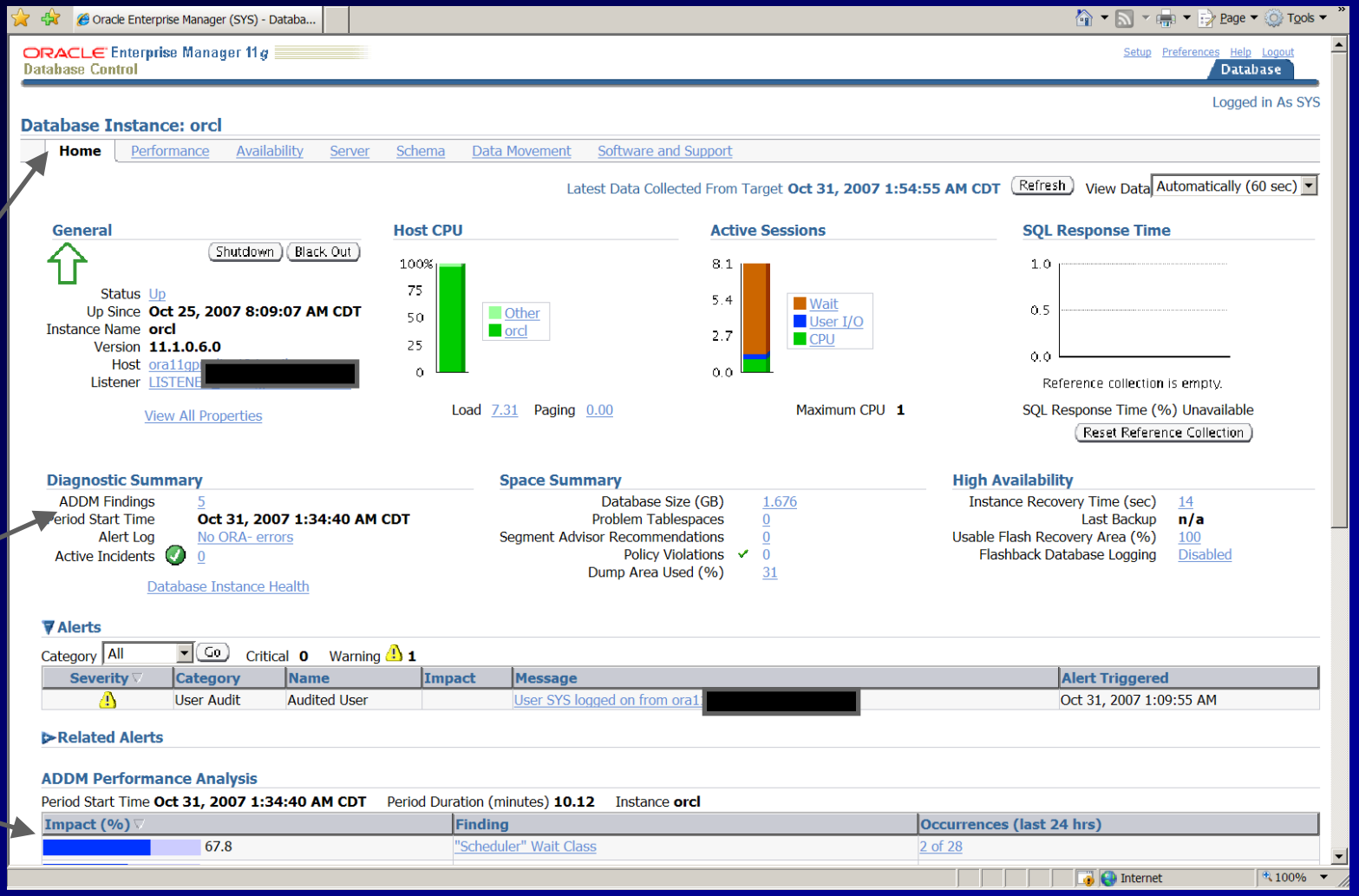

Oracle Enterprise Manager (SYSMAN) - Cluster Database: ioug - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://atlmdi8.us.oracle.com:7777/em/console/rac/racSitemap?type=rac_database&target=ioug&pageNum=5
Go Links
Cluster: IOUG >
Logged in As SYSTEM
Cluster Database: ioug
Latest Data Collected From Target Apr 23, 2006 11:19:22 AM EDT
Refresh
Home Performance Administration Maintenance Topology
Cluster Database topology presents the host view of a cluster database. Database instances, ASM instances, listeners, and interfaces information is available. You can optionally view configuration information. These views can also be used to launch various administration and configuration functions.
Show Only Hosts With Instances
Show Configuration Details
View Data Manually
Overview
Selection Details
Nothing Selected
Summary
Status Up
Up Instances 6 (⇧6)
Cluster IOUG
Alerts 5 8
Name: ioug_ioug1
Type: Database Instance
Host: atlmdi1.us.oracle.com
Critical Alerts: 1
Warning Alerts: 1
Status: Up
ASM: +ASM1_atlmdi1.us.oracle.com
Internet

# Enterprise Manager 11g
# Just a bit of changes...

Specific Database Instance

We have 5 ADDM Findings

Check them Here




# 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

# 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

1977 Government customer

2007

# 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)

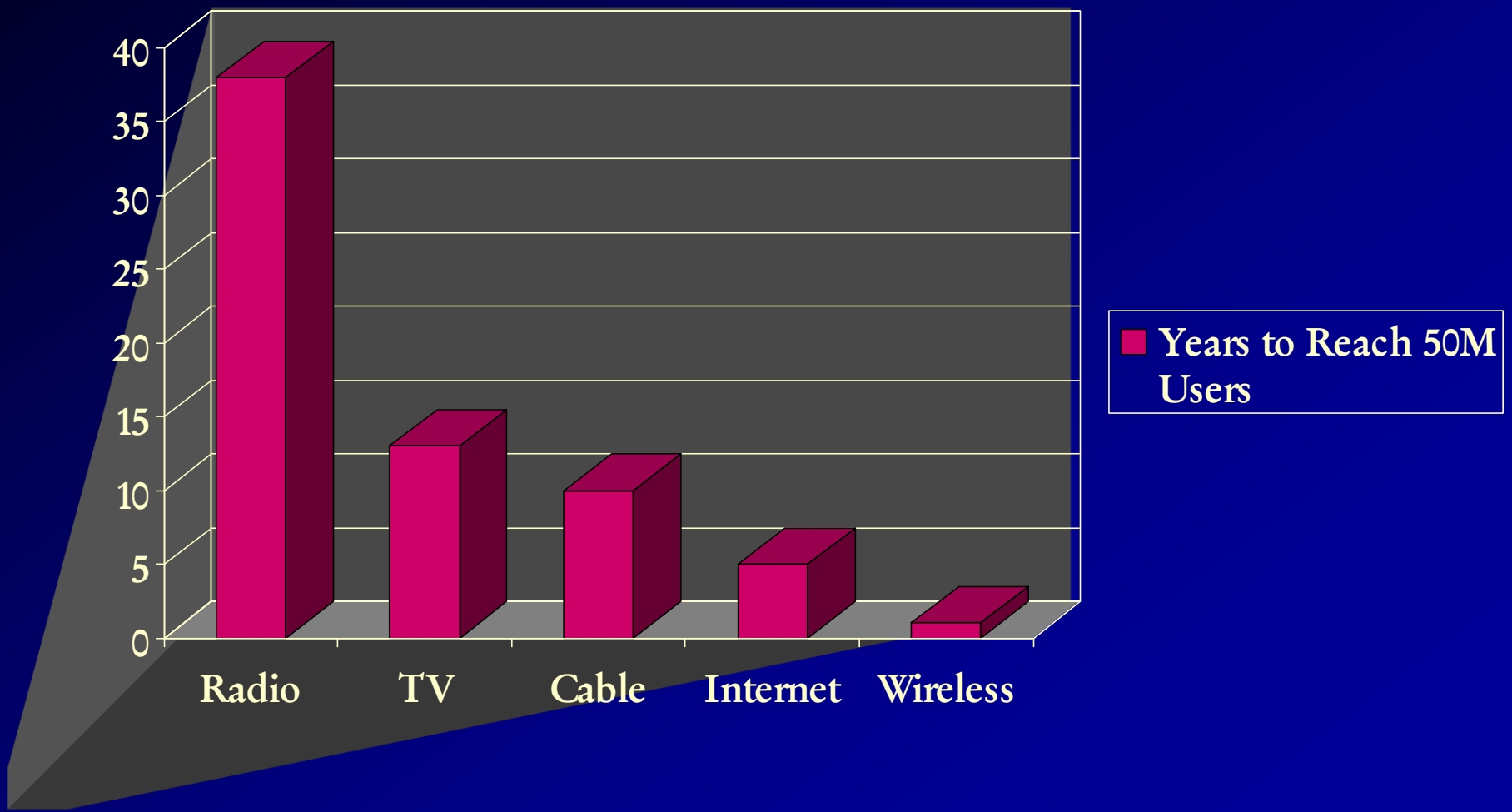
*or*

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

*or*

- 1 Mount Everest filled with Documents (approx.)

# Compelling Technology Statistics!

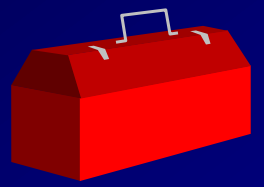


# Friedman's 6 Dimensions of Understanding Globalization*


- Politics (Merging)
- Culture (Still disparate)
- Technology (Merging/Merged)
- Finance (Merging/Merged)
- National security (Disparate)
- Ecology (Merging)

** Sited from Mark Hasson, PSU, Global Pricing and International Marketing.*

# V$ Views over the years



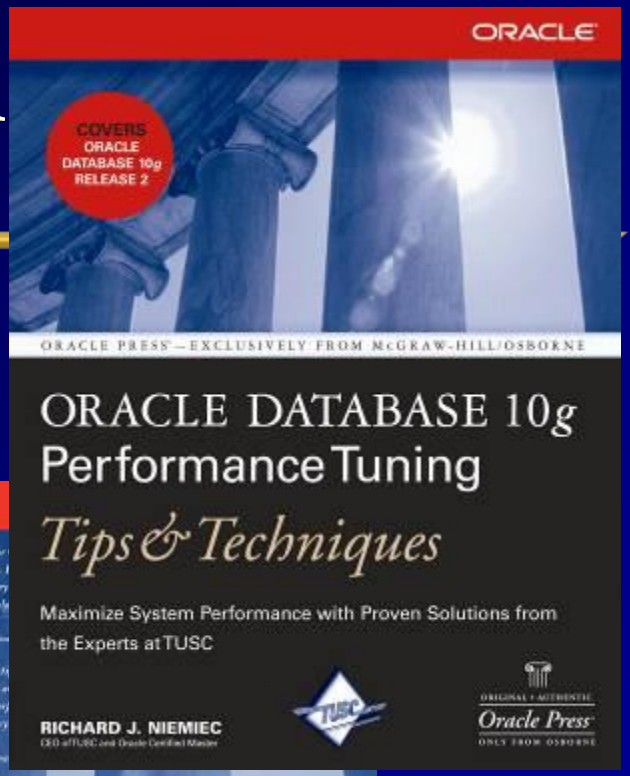
| Version | V$ Views | X$ Tables |
|---|---|---|
| 6 | 23 | ? (35) |
| 7 | 72 | 126 |
| 8.0 | 132 | 200 |
| 8.1 | 185 | 271 |
| 9.0 | 227 | 352 |
| 9.2 | 259 | 394 |
| 10.1.0.2 | 340 (+31%) | 543 (+38%) |
| 10.2.0.1 | 396 | 613 |
| **11.1.0.6** | **484 (+22%)** | **798 (+30%)** |

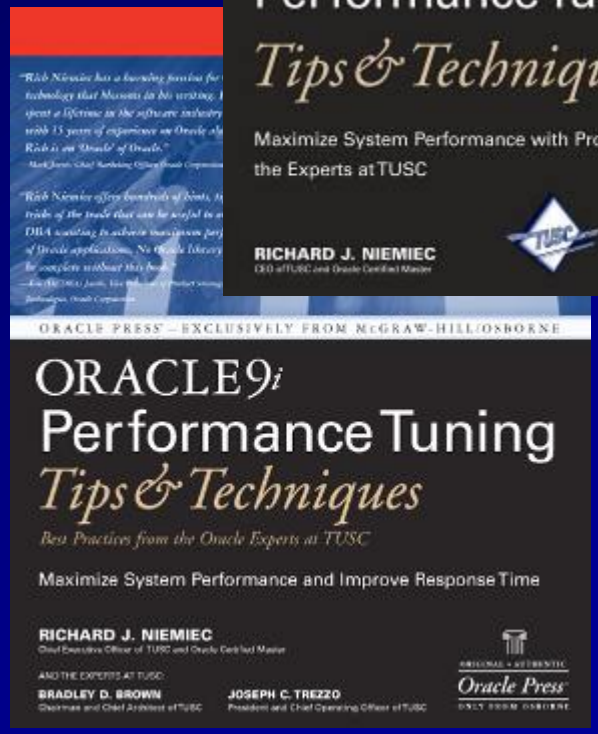
# Summary

- Start Me Up – Using Memory Target
- The Result Cache
- Invisible Indexes & Online Index Rebuilds
- Other Nice Developer Tools
- ADDM Enhancements
- SQL Plan Management & SQL Plan Baseline
- SQL Query Repair Advisor
- SQL Performance Analyzer
- Real Application Testing (Capture & Replay)
- Interval Partitioning & Partition Compressio
- DBA Tools and DBMS_STATS Enhancemen
- Grid Control & EM
- Security Enhancements & the Future Sizes
- Summary

# For More Information

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

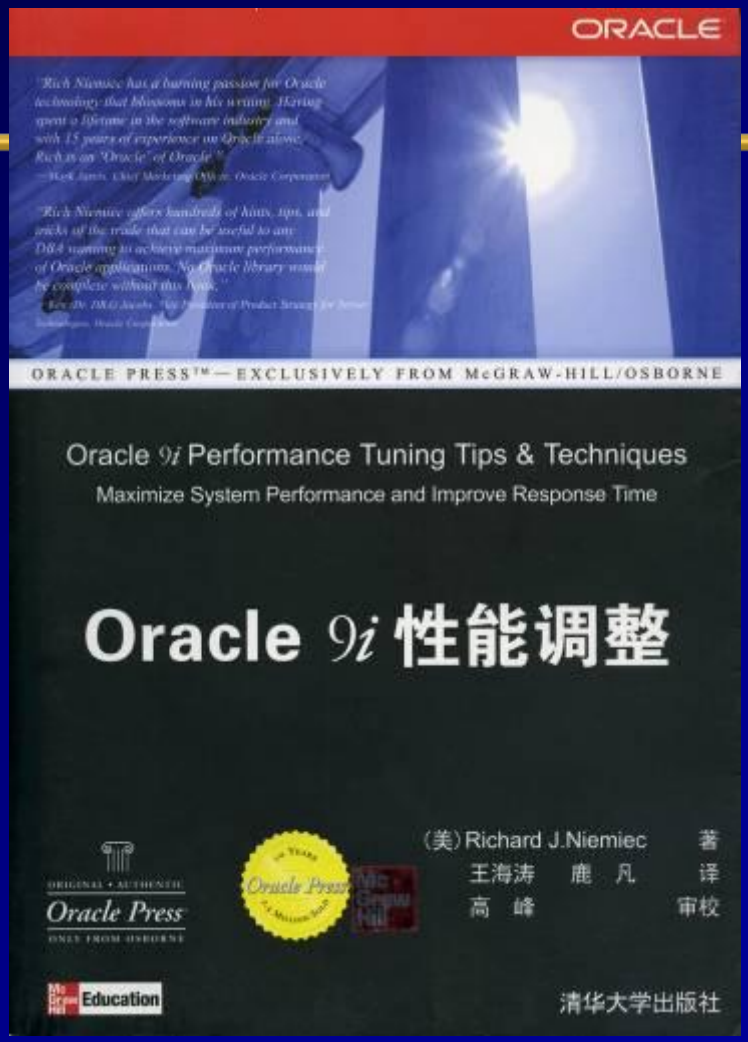



*"If you are going through hell, keep going" - Churchill*

# 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*

*"Life is not measured by the number of breaths we take, but by the moments that take our*

*-Dr.*

# Copyright Information

- Neither TUSC, Rolta, Oracle nor the author guarantee this document to be error-free. Please provide comments/questions to rich@tusc.com.
- 

Contact Information

Rich Niemiec: rich@tusc.com

**www.tusc.com**

# Save the Date!


May 3-7, 2009

Orange County Convention Center West; Orlando, Florida

# www.tusc.com



*"Success usually comes to those that are too busy to be looking for it."*

*- Henry David Thoreau*


# References


- www.tusc.com. www.roltatusc.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 & technet.oracle.com
- Thanks Dan M., Bob T., Brad, Joe, Heidi, Mike K., Debbie, Maria, Linda


- **Dedicated to the memory of Stan Yellott, Mark Beaton, Ray Mansfield**

# TUSC Services




- Oracle Technical Solutions
  - Full-Life Cycle Development Projects
  - Enterprise Architecture
  - Database Services
- Oracle Application Solutions
  - Oracle Applications & EMP Implementations/Upgrades
  - Oracle Applications & Hyperion Tuning
- Managed Services
  - 24x7x365 Remote Monitoring & Management
  - Functional & Technical Support
- Training & Mentoring
- Oracle Authorized Reseller

# TUSC Corporate Profile

- "The Oracle Experts" since 1988
  - Oracle Partner of the Year, 2002, 2004, 2007 & 2008
  - Editors Choice – Consultant of the Year 2002 & 2004
  - Authorship, User Groups and Various Awards
  - One of the first 6 Oracle Masters in the World
  - Certified Advantage Partner

ORACLE CERTIFIED ADVANTAGE PARTNER



# Rich's Overview (rich@tusc.com)


- President of TUSC – A Rolta Company:
  - Inc. 500 Company (Fastest Growing 500 Private Companies)
  - 7 Offices in the United States (U.S.); Based in Chicago
  - Oracle Advantage Partner in Tech & Applications
- Author (3 Oracle Best Sellers):
  - 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
- 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